1-1-2004

Contrast enhancement of ultrasound images using shunting inhibitory cellular neural networks

Murali M. Gogineni
Edith Cowan University

Edith Cowan University

Copyright Warning

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Contrast Enhancement of Ultrasound Images using Shunting Inhibitory Cellular Neural Networks

By

Murali Madan Mohan Gogineni

A thesis submitted for the degree of

Master of Engineering Science

at

School of Engineering and Mathematics

Edith Cowan University

Principal Supervisor: **Abdesselam Bouzerdoum**

July 2004

Abstract

Evolving from neuro-biological insights, neural network technology gives a computer system an amazing capacity to actually generate decisions dynamically. However, as the amount of data to be processed increases, there is a demand for developing new types of networks such as Cellular Neural Networks (CNN), to ease the computational burden without compromising the outcomes.

The objective of this thesis is to research the capability of Shunting Inhibitory Cellular Neural Networks (SICNN) to solve the clarity problems in ultrasound imaging. In this thesis, we begin by reviewing a number of traditional enhancement techniques and measures. Since the entire work of this thesis is based upon a particular model of the CNN, we present a brief review of CNN theory and its applications.

The SICNN biological inspiration, derivation and stability issues are reviewed with a view to understand its working principle. We then probe a general study of the feed-forward and recurrent SICNN systems. Here, the essential response properties of both SICNN systems are investigated in depth. The enhancing properties of the recurrent SICNN and its advantages compared to more traditional techniques are also studied. After a thorough investigation into the SICNN response properties, we introduce its application for enhancement in *Ultrasound Imaging* (UI) modality.

There are many techniques already available to us which perform high level enhancement. Unlike the regular problems encountered in images, the ultrasound images have some extra and unique obstacles to cross before they reach the same level of clarity as other diagnostic imaging systems. If not all, some of the problems associated with ultrasound imaging will be addressed in this thesis. Mainly, the reasons for the ambiguities in ultrasound detection are analysed and reasoned.

The next phase of this project mainly involves determining optimum SICNN parameters for the ultrasound image enhancement. Targeting the causes of the ultrasound ambiguities, an adaptive decay factor (a SICNN parameter), based on cell

intensity, is designed for selective enhancement. This decay factor controls the final impact of enhancement by allowing maximum enhancement rate for cells causing blurring and limiting the enhancement rate for high contrast cells.

An adaptively varying weight function to determine the appropriate neighbourhood effects is then investigated. In a given neighbourhood, this function is designed to generate the direction and magnitude of background effects based on the properties of the neighbouring cell intensities in comparison with the intensity of the cell under consideration.

Finally, the SICNN enhancement performance is evaluated on clinical ultrasound images and compared with those of conventional image enhancement techniques. The UI databases are collected from a wide variety of hospital equipments and contain both high and low quality images. The results of these experiments are quantified using a contrast measure, a contrast index, and dynamic range. Based on these results, we suggest a simple and effective method to improve the SICNN performance by exploiting the dynamic range of the ultrasound images. A net enhancement of approximately 25% has been achieved using this SICNN system.

Declaration

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education; and that to the best of my knowledge and belief it does not contain any material previously written by another person except where due reference is made in the text.

Signature: Date...12/10/04...

Acknowledgements

Firstly, I would like to thank my supervisor, Dr. Abdesselam Bouzerdoum. Without his guidance, support, constant probing, counter-questioning and most importantly his technical knowledge, this thesis would not have been completed. His cheerful personality and optimism have always been inspiring. In short, his esprit and wisdom make him the best mentor I ever had.

I would like to thank my father Dr. Kamalendra K. Gogineni for whose medical insights enriched my knowledge and understanding of diagnostic imaging, effectively easing my research strain.

I gratefully acknowledge the interest shown by the radiology departments of Royal Perth Hospital (WA) and MA Scan & Research Centres (India) in providing relevant data and information for this thesis.

I thank my colleagues and the departmental staff for their friendliness and their unforgettable effort in refining my research.

I thank my family, especially my grandparents, for their never-ending support and love through all the tyring and difficult periods of my studies and life. They provided me with everything I ever needed and always showed the right path to pursue my dreams.

My good friends Nirali Amin, Rhys Davis, Luke Osborne, Sairam Sudersanam, and Ravi Mariappa have always played an important role in initiating my future. I had many interesting experiences with them that could always be traced back to our friendship.

Finally, I thank God for giving me everything I cherish.

Abbreviations

1-D	1-Dimension
2-D	2-Dimensions
3-D	3-Dimensions
AHE	Adaptive Histogram Equalisation
BIBO	Bounded-Input Bounded-Output
CII	Contrast Improvement Index
CNN	Cellular Neural Networks
CT-CNN	Continuous Time Cellular Neural Networks
CY-CNN	Chua-Yang Cellular Neural Networks
DR	Dynamic Range
DT-CNN	Discrete Time Cellular Neural Networks
GEM	Gradient Enhancement Measures
GLE	Gradient-Like Enhancement
MNSCNN	Multiple Neighbourhood Size Cellular Neural Networks
NN	Neural Networks
NUP-CNN	Non-Uniform Processor Cellular Neural Networks
P-CNN	Polynomial Cellular Neural Networks
PDE	Partial Differential Equations
RCI	Relative Contrast Index
REE	Relative Edge Enhancement
RM	Ratio Memory
RMCNN	Ratio Memory Cellular Neural Networks
ROI	Region of Interest
SICNN	Shunting Inhibitory Cellular Neural Networks
UI	Ultrasound Imaging
UM	Unsharp Mask

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this introduction chapter, we start by presenting the motivation of this thesis and familiarizing the application that is being dealt with. In the next section, we highlight some of the targets set for this thesis. The research statement is stated in this section. As the application of Shunting Inhibitory Cellular Neural Networks (SICNN) for enhancing ultrasound images forms the main goal of this thesis, this application is given a brief introduction for understanding in the future chapters. Finally, we present the complete overview of this thesis.

1.1 Research Problem and Motivation

"Sound is for hearing and light for seeing things." This is the general convention used in life. The striking advance in science is the capability to adapt sound for visual reception. This unusual method is achieved by diverting ultra frequency sound waves into the required areas and recording the corresponding reflected wave patterns. This indirect method of visualizing images does not produce the same clarity as reflected light waves. This motivates the exploration of the enormous scope of development and enhancement in ultrasound imaging.

There are many techniques already available to us which perform high level enhancement. Unlike the regular problems encountered in images, the ultrasound images have some extra and unique obstacles to cross before they reach the same level of clarity as light images. If not all, some of the problems associated with ultrasound imaging will be addressed in this thesis. The required conditions to solve these problems and their reasoning are researched.

1.2 Research Statement and Goals

Before addressing the above cases, many traditional techniques would be explored and their strengths and weaknesses understood. Some of the measurement schemes are also listed for comparing different systems. Complex techniques using neural networks such as the Shunting Inhibitory Cellular Neural Networks (SICNN) are studied in depth. "***Adapting a SICNN to solve clarity problems in Ultrasound Imaging***" is the main goal of this thesis. The performance of such a SICNN and means to improvise its results will be discussed.

1.3 SICNN for Ultrasound Imaging

Lateral inhibition describes the complex mechanism by which sensory cells interact with each other, and was first proposed by Ernst Mach (1886) to describe the edge effects observed at the discontinuity between two different intensity bands. This phenomenon is now referred to as Mach bands. Since the pioneering work of Mach, inhibition has been shown to play an important part in the early visual processing system.

Multiplicative or shunting inhibition describes the case where the interaction between neighbouring cells is of a multiplicative nature; thus, it is inherently nonlinear. Pinter (1983a, 1983b) used lateral inhibition to explain the selectivity of visual units in the ventral nerve of insects for small objects, and also to explain the adaptation of the receptive spatial organization and the spatial modulation transfer function (*Pinter 1984, 1985*). Shunting inhibition has also found applications in image enhancement (*Jernigan and*

McLean, 1992; Bouzerdoum, 1994; Paradis and Jernigan, 1994), as well as in motion detection (*Bouzerdoum, 1991*).

Cellular neural networks (CNNs) were presented by Chua and Yang (1988) as a framework for analogue, nonlinear processing arrays. A CNN consists of a nonlinear processing node in a grid layout, with each cell being locally connected to its neighbouring cells. Many possible CNNs have been described, and they have found many applications in image processing, as CNNs have an excellent ability to process information locally, in both time and space.

The mammalian system consists of neurons in a grid-like structure, with many local interconnections which are used to interact with each other. The architecture and nonlinear processing ability of the CNN makes it ideal to duplicate the shunting inhibition in a mammalian system. Bouzerdoum and Pinter (1993) and Bouzerdoum (1994) were able to adapt and design a CNN to model such shunting inhibition.

Bouzerdoum (1994) successfully used these SICNNs to model aspects of primitive visual system. Iannella and Bouzerdoum (1996) synthesized the spatiotemporal receptive fields of the early mammalian visual system using a hierarchical model of the SICNNs. Pontecorvo and Bouzerdoum (1995, 1997) have also designed and applied SICNNs for edge detection. Cheung, Bouzerdoum and Newland (1999) have used these networks to investigate and enhance the dynamic range compression and contrast of colour images.

In this thesis, we take the SICNN developed by Bouzerdoum and Pinter (1993) and adapt it for enhancing medical images, ultrasound images in particular. The working principle and problems associated with ultrasound images will be discussed. The SICNN has various parameters which are tuned to dynamically vary its responses and solve the intricacies in ultrasound imaging. Furthermore, we experiment on some basic methods of enhancement and compare their performances to that of the SICNN. The networks are implemented on both high and low quality clinical ultrasound images. Using the test analysis, we suggest measures to improvise the SICNN performance and obtain consistent results for any given input image.

1.4 Thesis Overview

In Chapter 2 we present an overview of contrast enhancement theory and their applicability. We describe a number of intensity transformation and histogram processing techniques as found in literature. Contrast transformation methods are explained. Though filtering techniques are not the focus of this thesis, some of the major principles in this field are presented to give a comparative overlook of the systems and methods used in enhancement. Towards the end of this chapter, a few quantitative enhancement measures are explained for future use.

Chapter 3 begins by introducing the cellular neural network (CNN) theory and architecture. The dynamics of the CNN are summarised and analysed. The various types of CNNs available in literature, including the Shunting Inhibitory Cellular Neural Networks (SICNN) are introduced. Some of the popular applications of these CNNs are listed and briefed.

Chapter 4 covers the SICNN and its response properties. In this chapter, we first explain the basic concepts of linear and non-linear inhibition, followed by its biological design. We then use this biological design to electrically interpret the principle of shunting inhibition. Here we derive the general SICNN and discuss its stability issues. A general investigation into the classifications of SICNN systems, both feed-forward and recurrent type, is presented. We describe how the recurrent SICNN can be solved for the steady-state. The response properties of this recurrent SICNN are described for both uniform and step-edge inputs. We conclude this chapter by discussing the advantages of the SICNN and its performance comparisons to another technique, logarithmic enhancer.

Chapter 5 focuses on Ultrasound Imaging (UI) and the applicability of the SICNN to enhance UI images. We start by studying the principle of ultrasound and its application in the medical field. We also learn how the ultrasound scanners detect the tissues in a human body and how ambiguities in detection arise. The digital SICNN is then presented and its parameters described. We design these parameters to minimize the ambiguities in UI images while retaining the important contents of the images.

Chapter 6 deals with the performance analysis of the different networks that are tested. This chapter begins with an overview of the different types of UI images that are used for experimentation. We briefly present all the measures that will be used to quantify the experimental results. Performance analyses of some traditional techniques are also presented to compare the SICNN results. Based on these results, we then suggest a means to further refine the SICNN performance.

In Chapter 7, we present the conclusions of this thesis and a summary of its major contributions. Recommendations for future work are also provided.

Chapter 2

Enhancement and Measurement: A Review

2.1 Introduction

When capturing an image of any real-world scene, we can expect a number of degradations in the resulting image. These degradations may be due to the environment, such as poor lighting, or they may arise from inadequacies and limitations of the actual imaging device or the imaging technique. These degradations result in a direct reduction of the image quality.

Image enhancement is used to generate more visually pleasing and informative image. The enhancement techniques are generally problem oriented, as different applications have different needs. In this study, we are only interested in techniques for enhancing the contrast of an image.

In this chapter, we begin by presenting an overview of contrast enhancement theory in Section 2.2. Here intensity transformation techniques, histogram techniques, enhancement through filtering and contrast transformation methods are described.

Section 2.3 details contrast measurement methods used to quantify the enhancement in images. Of the many techniques, the contrast improvement index, gradient enhancement

measure and relative edge enhancement are explained as they have a significant role in Chapter 6 in reasoning a suitable measure for the outputs in this thesis.

2.2 Image Enhancement

Poor imaging environment and limited physical properties in analog scanners cause a definite loss in image information. However, such information drawbacks can be overcome by the use of digital enhancement techniques. We should understand that enhancement does not restore a degraded image to its original state; that is *image restoration.*

The principal objective of enhancement techniques is to process an image so that the originally acquired image is made more suitable for a specific application. This means that the adaptability of the techniques discussed is very much problem oriented. Thus, for example, a method that is used for enhancing medical images may not necessarily be the best approach for enhancing satellite images.

Enhancement techniques usually fall into two main categories: *spatial domain* methods and *frequency domain* methods. Spatial domain refers to the image plane itself, and the approaches in this category are based on direct manipulations of pixels in the image. In principle, processing in the frequency domain is totally based on making suitable modifications to the Fourier transform of the image under consideration. Under these two domains, there are many techniques and filters that can be used for enhancement. Out of all these, we are only interested in techniques that can be used to improve the contrast of an image.

Section 2.2.1 presents a general review of intensity transformation methods like contrast stretching, linear stretching and image negatives. Histogram processing techniques such as histogram equalisation and adaptive histogram equalisation are explained in Section 2.2.2. Though filtering techniques are not the focus of this thesis, the major principles in this field are presented in Section 2.2.3, to help us understand the various systems used in image enhancement. Contrast transformation methods are briefed in Section 2.2.4.

2.2.1 Intensity Transformations

The image processing functions in the spatial domain may be expressed as (*Gonzalez and Woods, 2001*):

$$g(i, j) = T[f(i, j)] \tag{2.1}$$

where $f(i,j)$ is the input image, $g(i,j)$ is the output image, and T is an operator defined over some neighbourhood of (i,j). Also, T can operate on a set of input images, such as performing the pixel-by-pixel sum of M images for noise reduction. The main approach of defining a neighbourhood about (i,j) is to use a square or rectangular sub-image area centred at (i,j) as shown in *Figure* 2.1.

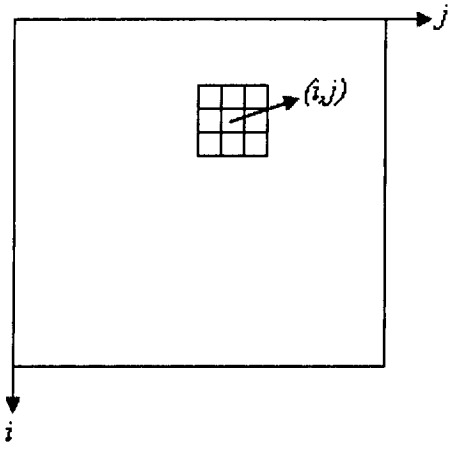


Figure 2.1 *A 3 x 3 neighbourhood about a point (i,j) in an image.*

The centre of the sub-image (mask) shifts from pixel to pixel starting, say, at the top left corner and applying the operator at each location (i,j) to yield g at that location. Although other neighbourhood shapes, such as approximations to a circle, are sometimes used, square and rectangular shapes are by far the most predominant because of their ease of implementation. The simplest form of T is when the neighbourhood is 1 x 1. In this case, g depends only on the value of f at (i,j) and T becomes a *gray level transformation* (also called *mapping*) of the form:

$$s = T(r) \tag{2.2}$$

where r and s denote the gray level of $f(i,j)$ and $g(i,j)$ at any point (i,j).

For example, if T (r) has the form shown in *Figure* 2.2(a), the effect of this transformation is to produce an image of higher contrast than the original by darkening

the levels below m and brightening the levels above m in the original image. In this technique, known as *contrast stretching*, the values of r below m are compressed by the transformation function into a narrow range of s towards black; the opposite effect takes place for values of r above m. In the limiting case shown in *Figure* 2.2(b), $T(r)$ produces a two level (binary) image. Simple, yet powerful, enhancement operations can be realized with gray-level transformation. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category are often referred to as *point-processing.*

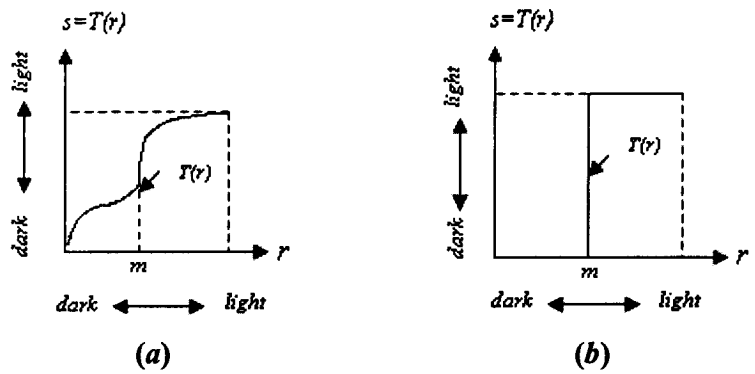


Figure 2.2 *Gray-level transformation functions for contrast enhancement (Gonzalez and Woods, 2001).*

2.2.1. A Contrast Stretching

An application of intensity transformation methods is the contrast stretching. Low-contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even setting of a lens aperture image acquisition. The idea of contrast stretching is to increase the dynamic range of the gray levels in the image. Significant contrast enhancement with considerable clarity in the image information can be achieved by applying this simple yet efficient technique to two dimensional images.

Figure 2.3(a) shows a typical transformation used for contrast stretching. The locations of points (r_1, s_1) and (r_2, s_2) control the shape of the transformation function. For instance, if $r_1 = s_1$ and $r_2 = s_2$, the transformation is a linear function that produces no changes in gray levels. If $r_1 = r_2$, $s_1 = 0$ and $s_2 = L-1$, the transformation becomes a *thresholding function* that creates a binary image. Intermediate values of (r_1, s_1) and (r_2, s_2) produce

various degrees of spread in the gray levels of the output image, thus affecting its contrast. In general, $r_1 < r_2$ and $s_1 < s_2$ is assumed so that the function is single valued and monotonically-increasing. This condition preserves the order of gray levels, thus preventing intensity distortion in the processed image.

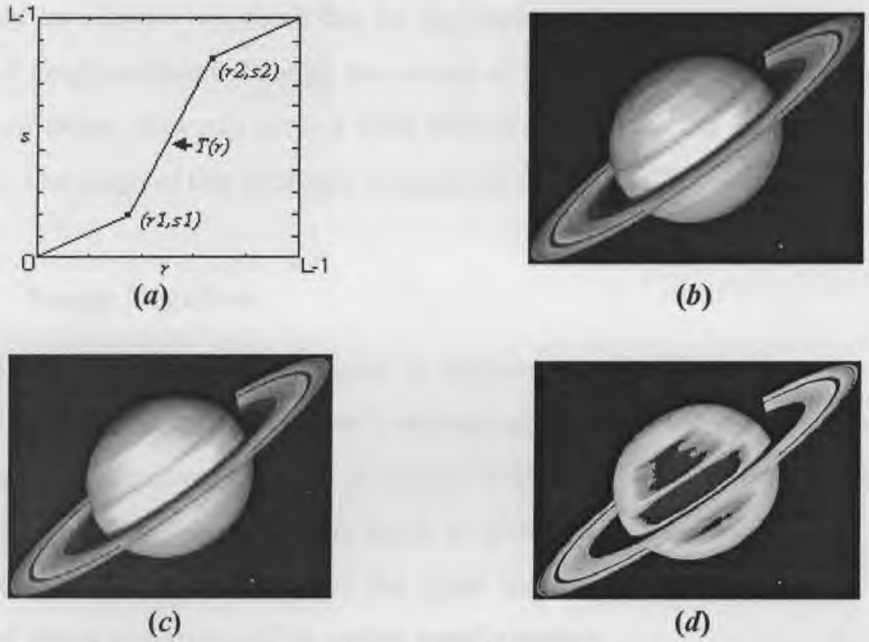


Figure 2.3 *Contrast stretching (a) form of transformation function (b) a low contrast image (c) after contrast stretching (d) after double-thresholding.*

Figure 2.3(*b*) shows an 8-bit image with low contrast, *Figure* 2.3(*c*) shows the result of contrast stretching, and *Figure* 2.3(*d*) shows the result of double-thresholding this image, with threshold levels of $r = 60$ and $r = 200$, the output is set to 256 (white) for any gray level in the input image of 200 or higher and 0 (black) for gray levels less than 60, and all other gray levels of the image are retained. A possible cross-over of gray levels at certain parts is experienced during simulation of results, as shown in *Figure* 2.3(*d*). This can be eliminated by pegging the results to the desired gray level as explained earlier.

2.2.1. B Linear Stretching

A slightly different approach is *linear stretching*, where the smallest input intensity in the original image is stretched down to a desired minimum I'_{min}, and the maximum input intensity is linearly scaled up to a desired maximum I'_{max}. Thus, the required

transformation for an input pixel of intensity I and for a desired output gray scale range $[I_{min}, I_{max}]$ is:

$$I' = I'_{\min} + \frac{(I - I_{\min})}{(I_{\max} - I_{\min})}(I'_{\max} - I'_{\min}) \tag{2.3}$$

Once again an adaptive approach can be implemented by processing the image over a small local neighbourhood. Though this is one of the traditional and simple techniques, the usage of neural networks gives a more effective output for the problem addressed in this thesis. The usage of this approach is examined further in *Chapter 5*.

2.2.1. C Image Negatives

Negatives of digital images are useful in numerous applications, such as displaying medical X-ray images, where a screen is photographed with a monochrome positive film and the resulting negatives are used as normal slides. The negative of a digital image is obtained by reversing the order from black to white so that the intensity of the output image decreases as the intensity of the input increases. *Figure*2.4 below shows an example of image negatives and its typical transformation.

(*a*) (*b*) (*c*)

Figure 2.4 *Image negatives* (*a*) *form of transformation function* (*b*) *image of a spine* (*c*) *negative of image.*

The transformation used for generating image negatives is probably one of the simplest, but yet has a very crucial application. Considering medical images in general, the white parts surrounded by dark regions form the most important information of the image. Thus, by reversing the intensities and then enhancing the edges, the sharpness of the image increases more efficiently yet retaining the vital information of the image.

2.2.2 Histogram Processing

The histogram of an image represents the relative frequencies of occurrence of gray levels in the image. Mathematically, the histogram of a digital image with gray levels in the range of [0, L-1] is a discrete function $h_c = n_c/N$, where N is the total number of image pixels, and n_c is the number of image pixels in the image with gray-level c; where c = 0, 1, 2,..., L-1. This function for all values of i provides a global description of the appearance of an image.

Before we look at some of the useful applications of histogram processing, it is important to understand how histograms are normally interpreted. The histogram in *Figure 2.5(a)* shows that the gray levels are concentrated towards the dark end of the gray-scale range. Thus this histogram corresponds to an image with overall dark characteristics. Similarly, as seen in *Figure* 2.5(*b*), a bright image has its gray levels towards the right of the histogram.

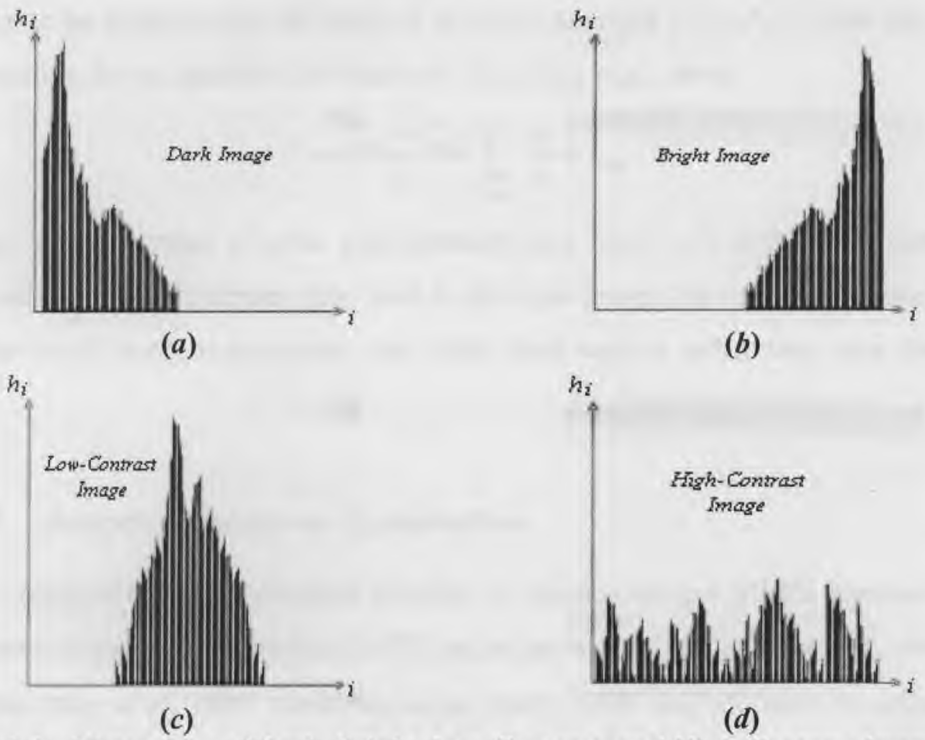


Figure 2.5 *Histograms of Dark, Bright, Low-Contrast, and High-Contrast Images.*

The histogram in *Figure* 2.5(*c*) has a narrow shape, which indicates a low dynamic range and thus corresponds to an image having low contrast. As all gray levels occur toward the

middle of the gray scale, the image would appear a murky gray. Finally, *Figure* 2.5(*d*) shows a histogram with a significant spread, corresponding to an image with high contrast.

Although histograms are global descriptions that say nothing specific about image content, the shape of the histogram of an image does give useful information about the possibility for contrast enhancement. By modifying the histogram of the image, the properties of the image can be varied. There could be various techniques to vary the contrast, but histogram equalisation is by far the most common one.

2.2.2. A Histogram Equalisation

Histogram equalization transforms the input image in such a way that the output image histogram is roughly uniform. Histogram equalisation tends to increase small contrasts, and reduce large contrasts. Let I' be the output after histogram processing, and the output histogram to be uniform over the range of desired intensities $[I'_{min}, I'_{max}]$, then the desired transformation for an input pixel of intensity I is (*Sonka et al., 1993*):

$$I' = \frac{I'_{\max} - I'_{\min}}{N} \sum_{i=I_{\min}}^{I} \frac{n_c}{N} + I_{\min} \tag{2.4}$$

where n_c is the number of cells with intensity gray level c, N is the total number of pixels, and I_{min} is the minimum gray level in the input image. An adaptive version of this technique would apply equalization over small local regions rather than over the entire image.

2.2.2. B Adaptive Histogram Equalisation

Intensity windowing was a common practice to enhance images till the introduction of the adaptive histogram equalisation (AHE) technique in the 1980's (*Pizer et al., 1984; Pizer et al., 1986; Pizer et al., 1987; Zimmerman et. al., 1988*). AHE was proposed to address the problems of display devices in depicting the full dynamic range in some medical images. Unlike intensity windowing, the AHE is automatic, reproducible, and sensitive to the local spatial information in an image. In this version of histogram equalisation, the

contrast enhancement mapping for a pixel is a function of an intensity region immediately surrounding the pixel. The intensity values in the region are used to calculate a histogram equalisation mapping, which is then applied to the pixel. More complex AHE schemes can be found in Paranjape et al. (1992).

2.2.3 Filtering

Though filtering techniques are not the focus of this thesis, it is important to understand filters as they would help us gain a comparative overlook of the systems and methods used in enhancement.

Filtering can be performed in both spatial domain and frequency domain. The spatial filtering schemes involve direct alteration of the gray levels of the pixels. For enhancement in the frequency domain, the Fourier transform of the original image is computed, and is multiplied by the filter's Fourier transform. The inverse Fourier transform of this product is calculated to obtain the enhanced image in the spatial domain.

In this section, common frequency domain filters such as high-pass filters and homomorphic filters are described. In the spatial domain, non-linear unsharp masking used for edge-enhancement is explained. The classical unsharp masking is also described.

2.2.3. A High-pass Filtering

In high pass filtering; only the high frequency components of the input signal, such as edges and noise are enhanced. To sharpen an image, the image's Fourier transform can be multiplied by the transform of a high-pass frequency filter such as the Butterworth high-pass filter (*Gonzalez and Woods, 2001*).

$$H(u,v) = \frac{1}{1+\left[\frac{w}{\sqrt{u^2+v^2}}\right]^{2n}} \tag{2.5}$$

where u, v are the frequency variables, w is the cutoff frequency, and n is the filter order. Again this approach emphasises the high frequency components (edges) of the image, which includes noise too.

2.2.3. B Homomorphic Filtering

Homographic filtering is a special case of a class of systems known as homomorphic systems. The motivation to this field of filters is to apply concepts and structures of abstract linear algebra to image processing. Among many others, homomorphic filtering has found applications in image restoration, speech processing and seismic signal processing. A typical homomorphic approach to image enhancement is illustrated in *Figure* 2.6 (*Oppenheim et al., 1968; Jernigan and McLean, 1992; Gonzalez and Woods, 2001*).

In homomorphic filtering, the input image $f(x,y)$ is considered to consist of two components - illumination $i(x,y)$, and reflectance $r(x,y)$; where

$$f(x,y) = i(x,y)r(x,y) \tag{2.6}$$

It is also assumed that the reflectance component changes drastically at intensity discontinuities, whereas the illumination component does not. Hence, the illumination component is associated with the slow changing background and the reflectance component is associated with high frequency edges. These components of the input image can be separated using the homomorphic filtering technique. The filter $H(u,v)$ is designed to affect the low and high frequency components differently. Typically the high-pass filter is used in order to strengthen edges. The natural logarithm allows us to apply classical linear image processing techniques to each component of the input.

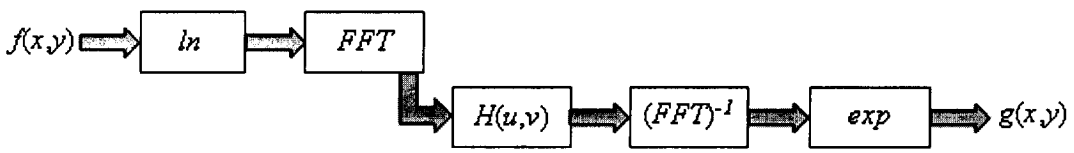


Figure 2.6 *Homomorphic filtering for image enhancement* (*Gonzalez and Woods, 2001*).

2.2.3. C Unsharp Masking

There exist many spatial filtering enhancement schemes such as mean and median filtering, but we shall not consider them here as they primarily remove noise and do not enhance edges. A simple but effective algorithm to enhance edges is known as *high-frequency boosting* or *unsharp masking* (*UM*) (*Gonzalez and Woods, 2001*), where the high-frequency contents of the input are partially added to the input. The enhanced output is:

$$\begin{aligned} I_{boost} &= \alpha.I - I_{LP} \\ &= (\alpha - 1).I + I - I_{LP} \\ &= (\alpha - 1).I + I_{HP} \end{aligned} \tag{2.7}$$

where I, I_{boost}, I_{LP} and I_{HP} are the original input, the boosted output, the low-pass and the high-pass versions of the input image, respectively. The parameter α controls the amount of the original image added to its high-passed version. A typical mask used to obtain the high pass image is:

$$\frac{1}{9}\begin{pmatrix} -1 & -1 & -1 \\ -1 & w & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

where, $w = 9\alpha - 1$, $\alpha \geq 1$. This technique relies on the fact that edges are usually of high frequency, hence this approach increases their strength compared to the background. Of course noise is also emphasised since it is of a high-frequency nature.

2.2.3. D Non-linear Unsharp Masking

As discussed above, the classical contrast enhancement filter, so-called *Unsharp Mask* (*UM*), increases the contrast by adding a high-pass version of the input signal to itself. Naturally, the edges are enhanced, but so is any noise present in the signal.

Guillon et al. (1996) developed a new class of adaptive non-linear contrast enhancing filters. These filters are called non-stationary filters because the filter mask depends on the local pixel values. The filter mask processes the pixel by a combination of high-pass and low-pass filter versions of the input.

Consider a filter mask M of size J x L centered on the pixel (i,k). Each coefficient m_{ik}^{jl} of this mask is viewed as a level of confidence of the current pixel belonging to the mask. Thus, $m_{ik}^{jl} \rightarrow 1$ for pixels to the center pixel and $m_{ik}^{jl} \rightarrow 0$ for others, and $M = \{m_{ik}^{jl} \in [0,1]$ and $(j,l) \in J \times L\}$.

Let I_{ik} be the intensity of the centre pixel of mask M, and I_{jl} be the intensity of pixel (j,l) in the mask. The algorithm needs a discriminating function that tends to one when the pixel values are similar, and tends to zero otherwise. A suitable function is

$$m_{ik}^{jl} = \exp\left[\frac{-(I_{jl} - I_{ik})^2}{2\sigma^2}\right] \tag{2.8}$$

where σ controls the width of the Gaussian curve. The mask M is computed for every pixel in the image. The proposed filter structure is shown in *Figure* 2.7.

The multiplier α is a weighted factor driving the contrast enhancement effect. The low and high pass versions of the input are computed as, respectively,

$$I_{ik}^{LP} = \frac{\sum_{(j,l)\in M} m_{ik}^{jl} I_{ik}}{\sum_{(j,l)\in M} m_{ik}^{jl}} \tag{2.9}$$

$$I_{ik}^{HP} = \sum_{(j,l)\in M} (m_{ik}^{jl} - \bar{m}_{ik}) I_{ik} \tag{2.10}$$

with $\bar{m}_{ik} = \frac{1}{JL} \sum_{(j,l)\in M} m_{ik}^{jl}$

This particular technique is called the Gradient-Like Enhancement (GLE) technique.

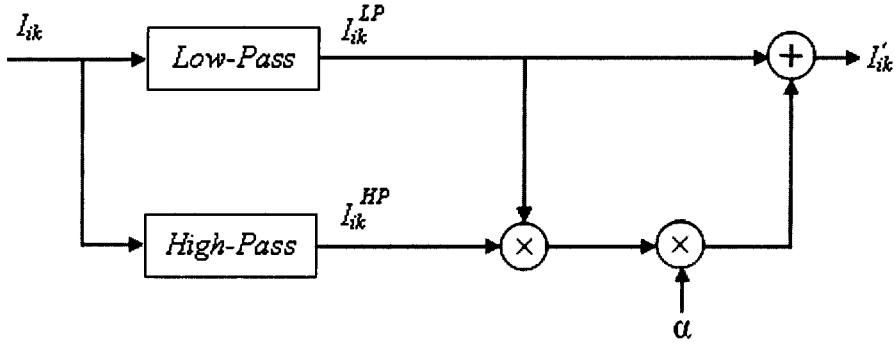


Figure 2.7 *Block diagram for Gradient-Like Enhancement (Guillon et al., 1996).*

The low-pass component I_{ik}^{LP} is a weighted mean over the mask M. The high-pass component I_{ik}^{HP} can be interpreted as local estimate of the gradient at that pixel.
The overall output of the system is:

$$I'_{ik} = I_{ik}^{LP}(1+\alpha I_{ik}^{HP}) \tag{2.11}$$

This technique attempts to improve the performance over the usual Unsharp Masking algorithm by multiplying the low-pass version and the high-pass version of the input, rather than multiplying the input and its high-pass version. Thus, edge enhancement can be achieved with this method, but with a reduced noise effect.

2.2.4 Contrast Transformation Methods

A number of schemes (*Gordon and Rangayyan, 1984; Dhawan et al., 1986: Beghdadi and LeNegrate, 1989: Dash and Chatterji, 1991*) aim to explicitly vary the local contrast of an edge in an image. All of the proposed methods begin by computing the local contrast in a small window, and then changing that contrast according to some function or transformation. The intensity of the central element of the local window is then recomputed according to the new contrast value for that pixel. Thus, the function or transformation determines the relationship between the output and input contrasts.

Based on Michelson's formula, the contrast of pixel (i,k) in a given local neighbourhood is defined as (*Peli, 1990*):

$$c_{ik} = \frac{|I_{ik} - I_0|}{|I_{ik} + I_0|} \tag{2.12}$$

where I_{ik} is the intensity of the pixel and I_0 is the local mean intensity.

Numerous transformations for this contrast have been proposed. Dhawan et al. (1986) investigated a number of transformation functions, including the tangent $\tan(nc_{ik})$, hyperbolic tangent $\tanh(nc_{ik})$, exponential $1-\exp(-nc_{ik})$, natural logarithm $\ln(1+nc_{ik})$, and the square root $\sqrt{c_{ik}}$ (n is a real scalar). These transformations map the contrast to the range of [0 1]. They not only increase the contrast but also increase the noise intensity, so the choice of the transformation function is usually a trade-off between the amount of

contrast enhancement and the allowable increase in noise intensity. For a contrast transformation $F(c_{ik})$, where $F(c_{ik}) \geq c_{ik}$ and $F(c_{ik}) \in [0, 1]$ for $c_{ik} \in [0, 1]$, the pixel's intensity is modified as

$$I'_{ik} = \begin{cases} I_0 \dfrac{(1+c'_{ik})}{(1-c'_{ik})} & \text{if } I_{ik} \leq I_0 \\ I_0 \dfrac{(1-c'_{ik})}{(1+c'_{ik})} & \text{otherwise} \end{cases} \tag{2.13}$$

where $c'_{ik} = F(c_{ik})$

2.3 Enhancement Measures

There exist a large number of image enhancement techniques, but only few methods that can quantify the amount of enhancement. To enumerate the effects in an enhanced image, defined measurements must be used.

Many researchers use visual inspection to compare different enhancement schemes. Visual inspection is a simple means of rating the change in the image quality, but is not entirely adequate for determining the enhancement performance. For example, humans are subjective-people, and may estimate the quality of an image in different ways.

We desire a means of rapidly and automatically measuring the improvement in an image with good consistency. For this reason, we shall consider quantitative measures of enhancement. Furthermore, the quantitative measures solely rely on the statistics of the image. The enhancement schemes considered here primarily enhance the contrast or edges, of the input image.

2.3.1 Contrast Improvement Index

Assuming that there is a step edge in a local neighbourhood, the contrast in a small window centered on pixel (i,j) is defined as (*Peli, 1990*):

$$c = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \tag{2.14}$$

where I_{max} is the maximum background intensity, and I_{min} is the minimum background intensity of the edge. This is also called Michelson's contrast.

It should be noted that *EQ* (2.14) measures the contrast in a small window centered on pixel (*i,j*), whereas *EQ* (2.12) gives the contrast of the actual pixel (*i,j*).

A method of evaluating contrast enhancement is to compare the contrast of the input image and the enhanced image (*Dash and Chatterji, 1991; Dhawan et al., 1986: Gordon and Rangayyan, 1984; Beghdadi and Le Negrate, 1989*). In 1994, Laine et al. defined the Contrast Improvement Index (CII) in a region of interest as:

$$CII = \frac{C_{out}}{C_{in}} \tag{2.15}$$

where $C_{out,}$ and C_{in} are the contrast at an edge in the output (enhanced) image and the original input image, respectively.

2.3.2 Gradient Enhancement Measure

Another possible image enhancement measure is the Gradient Enhancement Measure (GEM) as proposed by Harris (1997). This is simply a measure of the increase in the gradient of the edge after image enhancement.

2.3.3 Relative Edge Enhancement

Paradis and Jernigan (1994) used a measure called the Relative Edge Enhancement (REE) for 1-D step edges. This measure is primarily used when the intensity of the edge points increases relative to the background intensity. For enhancement of a step edge, as shown in *Figure* 2.8, the contrast of a pixel is defined as:

$$C_e = \frac{\Delta y_p}{\Delta y} \tag{2.16}$$

where Δy_p is the distance between the edge peaks, Δy is the difference between the background intensities, and c_e is the contrast of the edge.

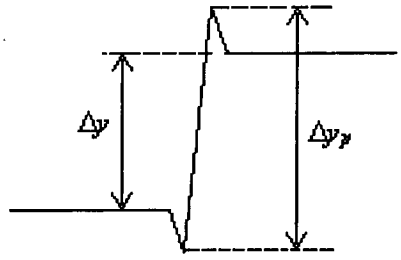


Figure 2.8 *A step edge whose edge pixels are enhanced relative to the background.*

The enhancement in an image compared to the original image (REE) can be written as:

$$REE = \frac{\Delta y_p / \Delta y}{\Delta x_p / \Delta x} \tag{2.17}$$

where Δy_p is the peak-to-peak edge variation, Δy is the background difference in the vicinity of the enhanced edge in the enhanced image, Δx_p and Δx are the corresponding measures for the original image.

2.4 Conclusion

In this chapter, we summarized and reviewed many important concepts that are to be used throughout this thesis. We examined in detail the contrast enhancing techniques using intensity transformations, histogram processing and contrast transformations. Filtering techniques in spatial and frequency domains were also explained.

Since the experimental comparisons in this thesis are mainly dependent on quantitative contrast measures, traditionally used schemes such as gradient enhancement measure, contrast improvement index and relative edge enhancement measure have been explained. The usefulness of these methods to quantify the enhancement in ultrasound images is later demonstrated in Chapter 6.

Chapter 3

Introducing Cellular Neural Networks

3.1 Introduction

Since its introduction by Chua and Yang in 1988, the Cellular Neural Network (CNN) architecture has proved to be one of the most widely applied neural network models. The CNN paradigm is a powerful framework for analogue nonlinear parallel processing arrays defined on a grid, featuring the local processing of cellular automata and the continuous dynamics of neural networks.

CNNs are suited to problems which are defined in space-time e.g., image processing tasks, and partial differential equations (PDEs).These problems are all characterized by fact that the information and interactions are generally constrained to small local areas, rather than large global ones. Thus, the main difference between CNNs and other Neural Network (NN) architecture is that in a CNN, all information is processed locally. However, global processing is still possible through dynamic diffusion of information.

The continuous dynamics and intense calculation capability of the CNNs with its local processing property makes them amenable to either electronic or optical implementations, which are usually difficult to achieve with the other forms of neural networks. If the

signals are continuous and or real-time operations are necessary, then the CNN is a good solution in terms of speed and time. The CNN model has also proven to be suited for VLSI implementations. CNNs have been implemented in VLSI chips capable of tera-flops operating frequencies (*Ecimovic and Wu, 2002*).

In this chapter, we first present an overview of the architecture and system dynamics of the general CNN in Section 3.2. We then look at a number of particular variants of this general model in Section 3.3. Here we also introduce the Shunting Inhibitory Cellular Neural Networks (SICNN), which will be used in this thesis. Finally, some of CNN's major applications are described in Section 3.4.

3.2 Cellular Neural Networks

The CNN architecture was proposed by Chua and Yang in 1988. CNNs are analog dynamic processors suitable for solving computational problems that can be formulated in terms of local interactions among signals placed on a regular structure (*Chua and Yang, 1988; Chua and Roska, 1993*). CNNs have already been applied to image processing problems such as filtering, edge detection, character recognition and object recognition. Due to the parallelism of the architecture, it can be applied to problems (such as video signal processing) where traditional methods cannot deliver fast throughput. There is much active research in the theory and implementation of CNNs and many applications of CNNs to real world problems have been reported.

CNN is a massive parallel computing paradigm defined in discrete N-dimensional spaces. Following the Chua-Yang definition, a CNN has the following properties (*http://www.ce.unipr.it/pardis/CNN/cnn.html# InterPoint*):

- A CNN consists of an N-dimensional regular array of elements (cells);
- The cell grid can be a planar array with rectangular, triangular or hexagonal geometry, a 2-D or 3-D torus, a 3-D finite array, or a 3-D sequence of 2-D arrays (layers);
- Cells are multiple-input single-output processors;

- A cell is characterized by an internal state variable, which may not be observable from outside the cell;
- More than one connection network can be present, with different neighbourhood sizes;
- A CNN dynamical system can operate in continuous (CT-CNN) or discrete time (DT-CNN);
- CNN data and parameters are typically continuous values;
- CNNs are recurrent networks; the final outputs typically require more than one iteration.

One of the main characteristics of the CNN is the localised connections between the cells. In fact, CNN differs from other NN mainly because, in a CNN, information is directly exchanged only between neighbouring cells. Of course this characteristic also allows global processing. Communications between non-directly connected units are possible through intermediate units. The CNN is considered as an evolution of Cellular Automata paradigm. Moreover, it was demonstrated that CNN paradigm is universal, being equivalent to the Turing Machine (*http://www.ce.unipr.it/pardis/CNN/cnn.html# InterPoint*).

In this section, we present an overview of the CNN architecture and operation, including some fundamental definitions and the most general equations defining its operation. This section also includes a discussion on the numerous types of cell grids possible, and how local interactions can cause a global flow of information throughout the network. We also briefly mention the stability issue of CNNs.

3.2.1 CNN Architecture

The basic unit of the CNN is referred to as a cell. In its electrical implementation, the cell generally contains linear and non-linear circuit elements, e.g. resistors and capacitors. Each cell is only connected to the cells in its local neighbourhood; hence only local interaction occurs.

CNNs can be defined over any dimension, though it is much easier to visualise them in 1-D or 2-D. In *Figure* 3.1 we show a 2-D CNN defined on a square grid, with each cell connected with only its immediate neighbours.

In a 2-D CNN with a total of M x N cells in M rows and N columns, let $C(i,j)$ denote the cell in the i^{th} row and j^{th} column. The r-neighbourhood of a cell $C(i,j)$ is the set of all cells within a distance of r from the cell (i,j) and is given by:

$$N_r(i,j) = \left\{ C(k,l) \middle| \max\left\{ |k-i|, |l-j| \right\} \le r, 1 \le k \le M, 1 \le l \le N \right\} \tag{3.1}$$

It is easy to show that this neighbourhood definition exhibits a symmetry property; that is, if $C(i,j)$ is a member of $N_r(k,l)$, then $C(k,l)$ is also a member of $N_r(i,j)$.

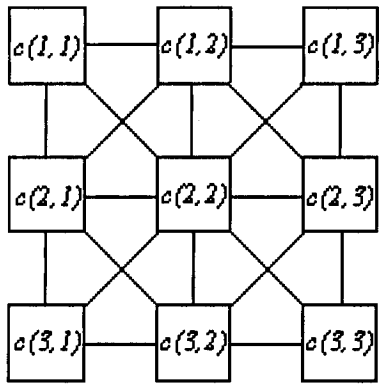


Figure 3.1 *2-D CNN defined over a 3 x 3 square lattice.*

The cell grid can be a 2-D array with rectangular, triangular or hexagonal geometry, a torus, or a 3-D array. Cells may be of the same type or belong to a different type. More than one connection network may also be present, each with different neighbourhood sizes – such as short range interaction and subsystem connections. The neighbourhood size may be as large as the network, in which case we have a *fully connected* network. Cellular networks, however, are usually implemented with only small neighbourhoods.

3.2.2 System Operation

The CNN is a dynamical system operating either in continuous or discrete time. A general form of the cell dynamical equations is defined as:

$$C\frac{dx_{ij}}{dt} = -ax_{ij} + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)y_{kl} + \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)u_{kl} + I \qquad (3.2)$$

$$y_{ij} = g(x_{ij})$$

where x and y denote the cell state and its output, respectively. A is the output feedback functional, B is the input functional controlling the effect of the neighbouring cells, u is the controlling input of the neighbourhood, g is the output functional of the cell and I is an independent bias.

Equation (3.2) represents the general form of state equations of CNNs. A number of implementations of these equations are discussed in the next section. In most of the cases, the system is non-markovian, i.e. the future network state depends also on its past states. *Figure* 3.2 depicts a block-scheme of a generic CNN iteration.

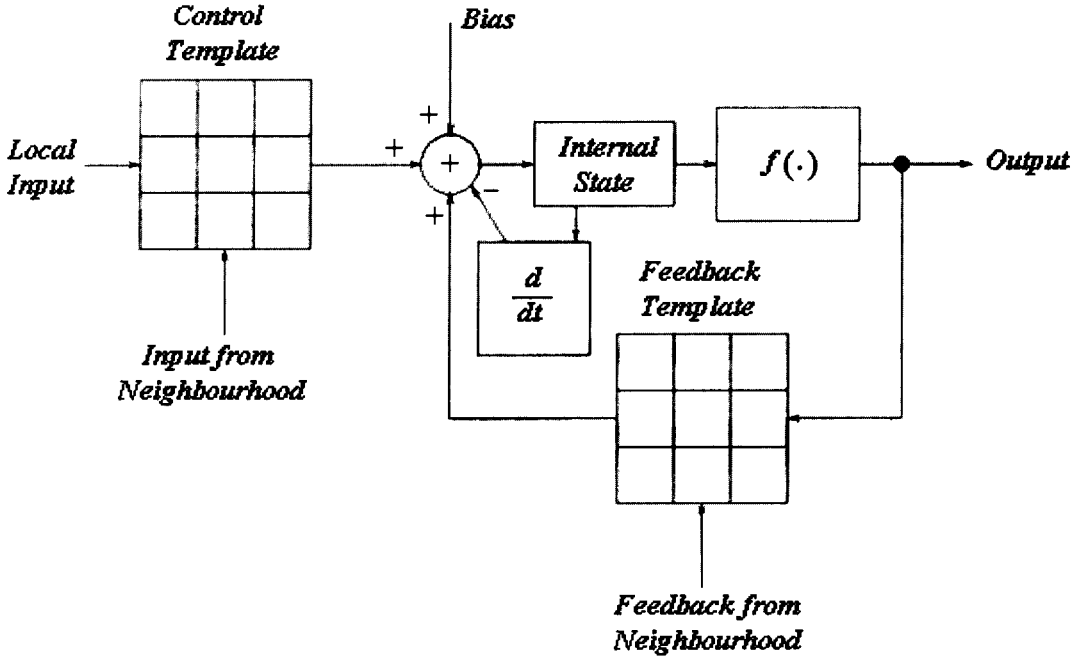


Figure 3.2 *Block-scheme of a generic CNN iteration, adapted from http://www.ce.unipr.it/pardis/CNN/cnn.html# InterPoint*

3.2.3 Stability

As with all dynamical systems, stability is an important issue with CNNs. For stability, the network must converge to a finite number of states. This can also be termed as complete stability. The stability of the Chua-Yang CNN (CYCNN) was widely investigated and a summary of the main results can be found in Civalleri and Gilli (1999). The general form of the CNN is not always stable, but stability can be proven for some subsets of the general model. The complete stability of the general CNN was studied in several papers (*Chua and Wu, 1992; Gilli, 1994; Arik and Tavsanoglu, 1996; Takahashi and Chua, 1998*), including the paper where the CNN paradigm was originally introduced (*Chua and Yang, 1988*).

3.3 Types of CNN

From such a broad and general definition of the CNN many variants are possible. As far as the dynamics is concerned, the CNN can be classified into two categories: stable CNNs and unstable CNNs. Generally, each variant is developed to suit a particular application. We shall review the variations of the general model such as the specific forms of the activation function, cell grid structure, template model, and discrete-time implementation. The variants discussed include polynomial or linear activation functions, uniform and non-uniform grid structures, and space-invariant and time-variant templates.

3.3.1 Polynomial CNN (P-CNN)

A variant of the general CNN is the *Polynomial* CNN (P-CNN) described by Barone et al. (1993), whose local feedback function is an odd-order polynomial. A third order polynomial function is shown in *Figure* 3.3. In the case of pattern recognition such functions act as shape attractors.

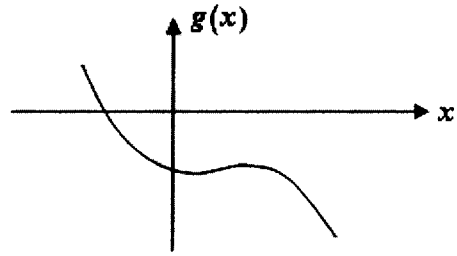


Figure 3.3 *A third order polynomial local feedback function.*

Recently new CNN models with polynomial interactions among the cells were introduced (*Tetzlaff et al., 1999; Laiho et al., 2000*). The use of higher order interactions has the advantage of solving computationally more complex problems, through an effective analog implementation (*Laiho et al., 2000*).

The polynomial interactions among cells alter the dynamics of the network, as they increase the number of distinct equilibria of the cell. Though the dynamics of such P-CNN's have not been deeply investigated, Corinto et al. (2002) present a comparative analysis of the P-CNN's stability with two other classifications of the CNN; the Chua-Yang CNN and the Full Range CNN. Corinto et al. (2002) showed that unlike the Chua-Yang and full range CNNs, the P-CNNs are stable under rather different conditions; in particular the symmetry of the template does not assure the stability of the network.

3.3.2 Non-linear, Delay Type and Non-Uniform Grid CNN

Rather than having two linear controlled sources $A(i,j\ ;\ k,l)\ y_{kl}$ and $B\ (i,j\ ;\ k,l)\ u_{kl}$ associated with cell $C(i,j)$ and neighbours $C(k,l)$, non-linear and delayed controlled sources can be employed (*Roska and Chua, 1992*), such as:

$$\hat{A}_{ij,kl}(y_{kl}+y_{ij})+A^{\tau}{}_{ij,kl}y_{kl}(t-\tau) \qquad \text{and} \qquad \hat{B}_{ij,kl}(u_{kl}+u_{ij})+B^{\tau}{}_{ij,kl}u_{kl}(t-\tau)$$

We can possibly have $\tau = \tau_{kl}$. The structure of the non-linearity is that it is at most a function of two variables: the output voltages of cell $C(i,j)$ and its neighbour $C(k,l)$.

Motivated by neurobiological structures, Roska and Chua (1992) introduced a Non-Uniform Processor CNN (NUP-CNN). An example of a NUP-CNN with two different processors (black and white) is shown in *Figure* 3.4. All inter-processor connections are space-invariant.

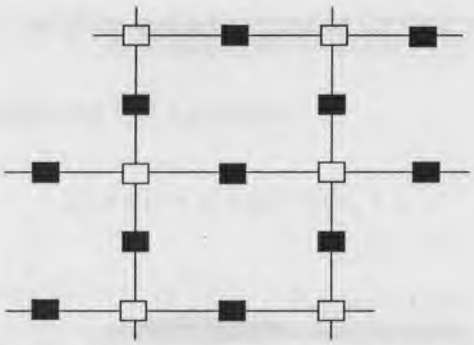


Figure 3.4 *Example of Non-Uniform Processor CNN (NUP-CNN). The black and white cells indicate two different types of processing units.*

CNNS can have several types of cell processor and more than one neighbourhood size, thus they are referred to as *Multiple Neighbourhood Size* CNNs (MNS-CNNs). *Figure*3.5 is a diagram of a MNS-CNN. Layer A (white cells) is a finely connected CNN with neighbourhood size r = 1. While layer B (dark cells) is a coarser grid with a neighbourhood size or r = 2 (connected to grid A). Only one cell of layer B is shown for simplicity.

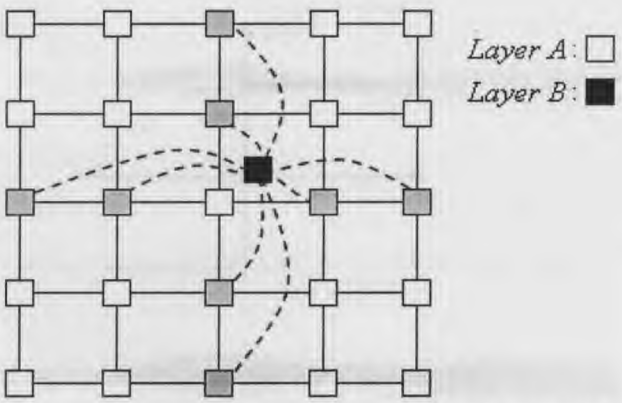


Figure 3.5 *Example of a Multiple Neighbourhood Size CNN (MNS-CNN). (White nodes indicate a finely connected CNN, and black nodes indicate a coarser grid. The r-neighbourhood for the white cells is 1, and 2 for the black ones.)*

3.3.3 Discrete-Time CNN (DT-CNN)

As opposed to continuous-time CNNs, discrete time CNNs (DT-CNNs) (*Harrer and Nossek, 1992*) have clocked variables and a comparator for their non-linear function.

They are defined by the following 1-D algorithm:

$$x_i(n+1) = a_i^j y_l(n) + b_i^j u_l + I_i \tag{3.3}$$

$$y_i(n) = f(x_i(n-1)) = \begin{cases} 1 & \text{if } x_i(n-1) \geq 0 \\ -1 & \text{if } x_i(n-1) < 0 \end{cases} \tag{3.4}$$

where i denotes the cell of interest, j is the cell in the neighbourhood of cell i, and n is the discrete time variable. The distinction from cellular automata is the continuously valued template coefficients and inputs.

This discrete-time recursive equation is called a discrete-time CNN (DTCNN). If $f(.)$ is not the standard nonlinear function, but is so-called the hard-limiter $f_h(.)$ with the limits as shown above (also see *Figure* 3.6) (*Chua and Roska ,2002*).

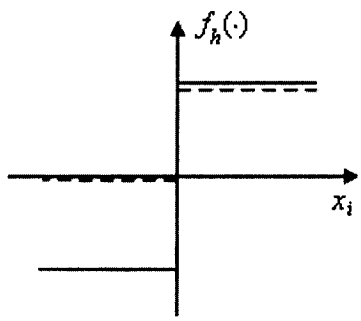


Figure 3.6 *Graphical representation of the hard limiter $f_h(.)$*

In continuous time CNNs, the propagation speed depends upon the derivative of x_i, which in turn depends upon the template and input/output signals. There are several different physical implementations for DTCNN including software, digital hardware, and special

purpose VLSI. Advantages of DT-CNNs include constant propagation time, simpler simulation (no numerical integration required), and *insensitivity* to template coefficients if they are chosen appropriately.

3.3.4 Time-Variant Template DT-CNN

A more generalized architecture for DT-CNN is the extension to time-variant templates (*Harrer, 1993*). A template is normally a matrix with numerical values describing the amount of interaction between neighbouring cells. Time variant DT-CNNS have cyclic templates; that is, templates whose coefficients are changed at every iteration step, and with the entire set of templates applied periodically in a cyclic manner. With such a paradigm, the hardware can be reduced; hence, the realization can be simplified. Applications of time variant template DT-CNNs include skeletonisation and half-toning (*Crounse et al., 1993*).

3.3.5 Shunting Inhibitory CNN (SICNN)

The architecture and nonlinear processing ability of the CNN makes it ideal for modelling nonlinear inhibition in the mammalian system which typically consists of neurons in a grid-like structure, with many local interconnections. Bouzerdoum and Pinter (1993) and Bouzerdoum (1994) were able to design a CNN to model early processing in the mammalian visual system. Using a hierarchical model of such CNNs, Bouzerdoum (1994) was able to synthesize both the centre-surround receptive field of retinal ganglion cells, and the orientation selective receptive of cortical cells, in the ON- and OFF-channels of the parvocellular system. Iannella and Bouzerdoum (1996) used a hierarchical network of SICNN to synthesize the spatiotemporal receptive fields of the early mammalian visual system. This CNN architecture will be discussed in detail in the next chapter since our edge enhancement operator will be based on the SICNN.

3.4 Applications of CNN

The mathematical model of a CNN consists of a large set of coupled nonlinear differential equations. This knowledge of the dynamic behaviour is essential for developing rigorous design methods for establishing new applications. Many CNNs have been proposed, each designed with a particular task in mind.

Since their invention, CNN's – homogeneous arrays of identical and identically coupled cells- have been intensely investigated for their applications in fast image processing, particularly where local processing of information is either necessary or advantageous. Significant and successful applications of CNNs include, but not limited to, the following:

- ***Feature extraction***: Slot (1992) demonstrated the application of CNNs to binary image processing. The user specifies the feedback and feedforward operators depending upon the desired features to be extracted, and the CNN is then able to reconstruct an output which is a modified version of the input, with the desired outputs emphasized in greater detail.

- ***Character recognition***: Sziranya and Csicsvari (1993), and Suzuki et al. (1992) employ CNNs to extract the necessary features of the input, which can then be used by a classification network to identify handwritten characters. The recognition rates for both networks were around 90%, with extremely fast recognition speeds due to the inherent parallelism. Sziranya and Csicsvari's system was able to identify 100,000 characters/second with a recognition rate of 95% when implemented in hardware.

- ***Motion detection***: Cimagalli et al. (1993) detect the trajectory of moving objects in a real-time noisy environment. Roska et al. (1992) define various templates to detect different types of motion, where the discrete-time inputs are fed into the network, and the resulting steady-state outputs give the necessary information for estimating the direction and magnitude of the velocity vector. For more applications of CNNs in this area, readers are referred to Shi et al. (1993).

- ***Spatial recognition***: Perez-Munuzuri et al. (1993) use a CYCNN to implement spatial recognition, i.e., recognizing open curves from closed ones, and locating the shortest path between the two locations.

- ***Logical Boolean functions***: Galias (1993) employs a time varying template CNN to define an arbitrary boolean function on the *r*-neighbourhood.

- ***Half-toning***: Crounse et al. (1993) were able to reproduce more faithful binary reproductions of the original image with a CNN than those produced by error diffusion, a standard algorithm for half-toning.

- ***Mathematical simulations***: Mathematical calculations can generate solutions to many problems. But considering the various conditions and experimenting on the huge number of possibilities was always considered to be a difficult task. Comparatively speaking, this problem has been eased by Roska et al. (1995), whose CNN could solve such rigorous partial differential equations and simulate nonlinear waves.

- ***Chip designs***: CNNs have found wide applicability in many hardware applications, many of which have been realized. Now CNNs have applications in the designing of general function chips. One of the successful designs would be by Linan et al. (2000), who designed an analog input/output 64x64 CNN Universal Machine Chip Prototype with 7-bit Analog Accuracy.

- ***Pattern formation***: An interesting phenomenon which has been shown to appear in CNNs is that of Pattern formation (*Goras et al., 1995a, 1995b, 1995c; Crounse et al., 1995*). Using a decoupling technique, Goras et al. (2002) showed that patterns can be produced by the input when the CNN is stable and by both the input and the state when at least one spatial mode is unstable.

- ***Ratio memory and Pattern Recognition***: CNNs application as neural associate memories for pattern learning, recognition and association has been demonstrated in great detail (*Liu and Michel, 1993; Lukianiuk, 1996; Kawabata et al. 1997*). The ratio

memory (RM) of Grossberg outstar structure, has been incorporated in neural networks for memory associated image processing (*Lan and Wu, 1995; Lan and Wu, 1995; Wu and Cheng, 1997*). The RM has also been incorporated in the CNN (RMCCN) and has been used for pattern learning and recognition (*Wu and Cheng, 2000*). Cheng et al. (2002) developed a new type of RMCNN with spatial-dependent self-feedback weights for an enhanced storage capacity and better pattern recognition from noisy images.

- Other applications include Hole-filler (*Matsumoto et al., 1990a*); Shadow-detector (*Matsumoto et al., 1990d*); Image thinning (*Matsumoto et al., 1990b*); Connected component detector (*Matsumoto et al., 1990c; Cruz and Chua, 1991*), Sensors/processors for multimedia applications (*Sheu et al., 1998*), etc.

3.5 Conclusion

In this chapter, we have presented a review of CNN theory. We began this chapter with an overview of the general CNN architecture and system operation. The stability issues were also described in this section. We then described five variants of the general CNN. Here the Shunting Inhibitory CNN was also introduced and the prominent works using this model were listed. In the final section of this thesis, the important applications of these CNN models were concisely summarised.

Chapter 4

Shunting Inhibitory Cellular Neural Networks: Network Architecture and Properties

4.1 Introduction

This chapter presents a detailed analysis of the SICNN systems, which includes the response properties of both feed-forward and recurrent SICNNs. We begin by detailing the concepts of linear and non-linear lateral inhibition in Section 4.2. As the SICNN system is derived from biological insights, we present a brief discussion of the biological shunting inhibition in Section 4.3. The working principle of a biological neuron is used to achieve this.

Based on working principle of the biological neuron, an equivalent electrical circuit showing inhibition is explained in Section 4.4. In this section, the derivation and stability issues of the SICNN drawn from such a circuit are reviewed in detail. Section 4.5 presents two classifications of the SICNN - feedforward and recurrent. The state analysis and response properties of both these systems are discussed. The feedforward system's dynamics, responses to step-edge inputs, and responses to asymmetrical and symmetrical weights are detailed.

The recurrent SICNN is used as the main network in this thesis. Hence, the steady state response, convergence properties and step edge response properties are studied in detail.

The response properties of the recurrent SICNN are compared with those of the feedforward SICNN, and the reason for our selection of the recurrent system is given.

Finally in Section 4.6, the advantages of the SICNN for image enhancement are mentioned. The dynamic range compression property of the SICNN is compared with other commonly used systems, like the logarithmic system.

4.2 Linear and Non-Linear Lateral Inhibition

In this section we begin by understanding two commonly used terms, luminance and brightness. This is followed by the concept of lateral inhibition and the roles of linear and non-linear inhibition.

Luminance is a measure of the intensity of light energy emitted by a luminous surface. But, the brightness of an object is defined as a measure of the light perceived by an observer. For example, to an observer, the brightness of an infra-red source could be almost zero even though the source is emitting energy. The brightness of an object also depends on the background of that object. Consequently, two objects can have the same luminance but different brightness depending upon their surroundings. This effect gives rise to concepts such as simultaneous contrast and Mach bands (*Mach, 1886*).

Lateral inhibition is often found in the preliminary stages of sensory processes such as touch and vision (*Deutsch and Deutsch, 1992*). It is a concept that explains information sharing between neighbouring sensory nerve cells. Linear lateral inhibition was first proposed by Ernst Mach to explain the border contrast effects commonly referred to as Mach bands (*Mach, 1886a; Mach, 1886b*). *Fig.* 4.1 shows an example of Mach bands at the transitions between regions of different intensities. Mach bands are the illusory dark and light bands on either side of each transition. This phenomenon is essentially due to inhibition causing the perceived brightness to differ from the actual luminance. *Fig.* 4.1(*c*) shows a receptive field similar to the one proposed by Mach, where the under and overshooting of brightness at transition borders is visibly shown.

referred to as shunting inhibition. Since the nonlinear lateral inhibition forms the base of this thesis, the mathematical details are discussed in depth in the following sections.

Nonlinear, or multiplicative lateral inhibition has been used to explain important theories such as; selectivity of visual units in the ventral nerve of insects (*Pinter, 1983a; 1983b*), and adaptation of the receptive field spatial organization and the spatial modulation transfer function (*Pinter, 1984; 1985*). Recently, nonlinear lateral inhibitory neural networks have also found application in image processing, mainly for image enhancement (*Jernigan and McLean, 1992; Bouzerdoum, 1994; Paradis and Jernigan, 1994; Cheung 1999; Chua and Roska, 2002*).

4.3 Biological Neuron

Insect and mammalian visual systems have long been a constant source of inspiration for computer vision researches. The ability of these visual systems to successfully operate under a wide range of conditions has drawn admiration from neurobiologists, computer scientists, and engineers alike.

Taking image resolution and edge-detection as examples, both mammalian and insect visual systems are able to process the vast amounts of incoming data in terms of edges, as well as other primitives (*Marr and Hildreth, 1980*).To detect the distribution and intensities of the incoming light, the surface of the retina in the eye is covered with discrete light receptors called *cones* and *rods*. Each cone is connected to only one nerve ending, hence resulting in high resolution and is responsible for colour vision. Majority of the receptors in the eye are the rods and many rods together are connected to a single nerve ending. This reduces their resolution but increases their sensitivity to low-levels of illumination. Electrical impulses from all the nerve endings exit the eyeball through the *Optic nerve*.

The *neuron* is the fundamental processing unit in the human nervous system. Over twelve billion nerve cells or neurons in the brain communicate with millions more in the body. A nerve signal is received by the neuron's *dendrites* which are connected to the *cell body* or *soma* and travels along the *axon*, a thin tube up to three feet long. *Figure* 4.2 shows the structure of a basic neuron with its most important connections.

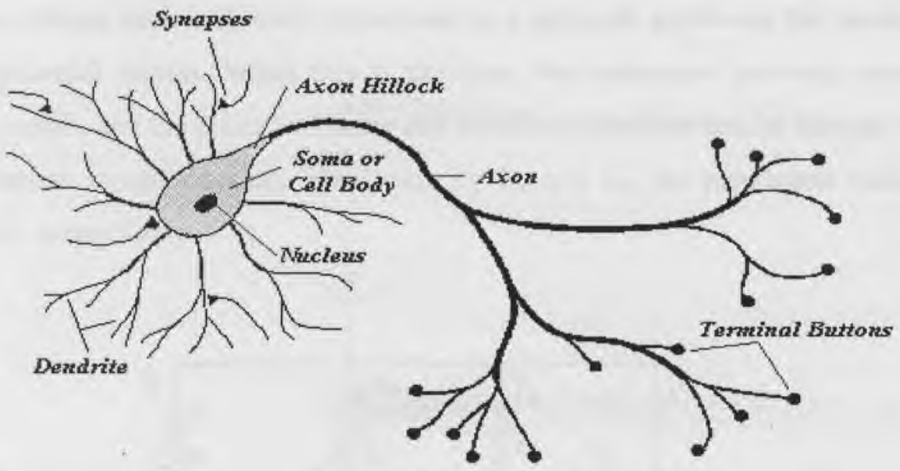


Figure 4.2 *Schematic drawing of a typical neuron adapted from http://vv.carleton.ca/~neil/neural/neuron-a.html*

The axon maintains a chemical balance with more potassium ions inside and more sodium ions outside. When a signal is transmitted, the *myelin sheet* (not shown in figure) covering the axon allows the different ions to leak through. The K^+ and Na^+ ions change places creating an electric signal that travels along the membrane. The gap between two neurons is called the *synapses*. The synapses could be of the inhibitory or excitatory types. When the impulse reaches the synapse, chemical transmitters are discharged which discharge the impulse to the next neuron. The first neuron returns to a resting state, while the different ions begin to move in the second neuron. Thus the impulses are passed on till they reach the required destination.

4.4 Electrical Interpretation of Shunting Inhibition

In this section we describe an electrical representation of the biological neuron. We discuss the working of this electrical circuit in comparison with the basic working of the biological neuron. This electrical circuit is further simplified by mathematically deriving a generalized form of the SICNN and concluding on its stability analysis.

An equivalent electrical circuit for a biological neuron is shown in *Figure* 4.3 (*Bouzerdoum and Pinter, 1993*). In our discussion we shall call a path from the synapses to the soma a *channel*. This circuit can be seen as an approximation to a uniform lump of

membrane, where each cell may correspond to a sub-unit which on the whole has a uniform potential within. When this is the case, the resistance between synapses is negligibly small, and the entire excitatory and inhibitory synapses can be lumped together into a common circuit element. We denote by V_m and C_m the membrane voltage and capacitance, respectively.

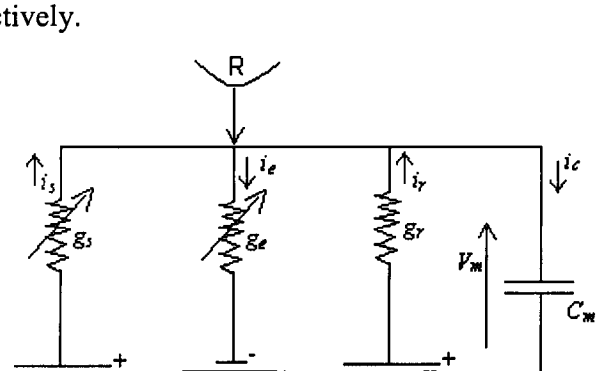


Figure 4.3 *An equivalent electrical representation of a biological cell, or neuron.*

When the neuron is not excited, or at rest, the corresponding resting conductance and batteries of all possible channels can be lumped together into the resting conductance g_r and resting potential V_r. Thus, the conductances of all excitatory and inhibitory channels are zero. By activating the excitatory synapses, the conductance of the corresponding ionic channels increases. This conductance of the ionic channels, also known as membrane conductance, causes the sodium N_a^+ ions to enter the soma. This flow of ions is represented by a modulation of the conductance g_e, with the polarity of the potential V_e, which reflects the direction of ionic flow. In the actual fact V_e is a measure of the flow of ionic charge across the channel.

The inhibitory synapses are assumed to be of the shunting type since this branch of the circuit, consisting of g_s and V_s, shunts the rest of the circuit. Upon excitation of these inhibitory synapses, more chloride Cl^- ions enter the soma through their respective ionic channels. This also increases the outer membrane potential with respect to that of the inner membrane. The voltage V_s represents the flow of Chloride ions across the channel. The changing Cl^- ionic channels can be represented electrically as a modulation of the conductance g_s. Though the Cl^- ions are dominant in causing shunting inhibition, the

ionic channel for potassium K^+ ions also plays a considerable role in exerting inhibition by leaving the soma and increasing the outer membrane potential.

Due to the inherent potential difference across the membrane, there is always a constant diffusion of charge across it. At the *steady-state*, equal amounts of charge flow in and out to the soma (represented by the inhibitory and excitatory potentials in the circuit). Thus when the cell is excited, the excitatory conductance g_e becomes larger, indicating an excitatory effect, and g_s increases to divert current from i_r to i_s. This diversion causes the circuit to clamp the cell to its resting potential, in other words, it attempts to maintain a constant equilibrium voltage, i.e $V_s = V_r$. Electrically this is referred to as inhibition.

4.4.1 Derivation of SICNN

The equivalent electrical circuit of a simplified biological cell, or neuron, is shown in *Figure* 4.3. KCL for this circuit can be written as

$$i_s + i_r = i_e + i_m \tag{4.1}$$

As we know that $i=Vg$, where i is the current, V is the voltage and g is the conductance the current equations can be written as

$$i_e = g_e(V_e + V_m) \qquad i_r = g_r(V_r - V_m) \qquad i_s = g_s(V_s - V_m) \qquad i_m = C_m \frac{dV_m}{dt}$$

By substituting the current equations in Equation (4.1) we get the nodal equation for the circuit (*Bouzerdoum, 1991*):

$$C_m \frac{dV_m}{dt} + g_e(V_e + V_m) - g_r(V_r - V_m) - g_s(V_s - V_m) = 0 \tag{4.2}$$

where i_e is the excitatory current; V_r and g_r are the lumped resting potential and resting conductance, respectively; V_s and g_s are the lumped synaptic battery and synaptic conductance of the inhibitory channels; C_m and V_m are the membrane capacitance and membrane voltage, respectively.

Let ΔV be the deviation of the membrane voltage from the resting potential, i.e. $\Delta V = (V_r - V_m)$, and with $V_s = V_r$, the change with time of V_m relative to V_r is described by the differential equation:

$$C_m \frac{d}{dt}(V_r - \Delta V) = g_r(\Delta V) + g_s(\Delta V) - g_e(V_e + V_m) \tag{4.3}$$

$$\frac{dV_r}{dt} - \frac{d}{dt}\Delta V = \frac{g_r}{C_m}(\Delta V) + \frac{g_s}{C_m}(\Delta V) - \frac{g_e}{C_m}(V_e + V_m) \tag{4.4}$$

But dV_r/dt can be equated to zero because V_r is a constant resting potential.

$$-\frac{d}{dt}\Delta V = \frac{g_r}{C_m}(\Delta V) + \frac{g_s}{C_m}(\Delta V) - \frac{g_e}{C_m}(V_e + V_m) \tag{4.5}$$

Since the resting potential is always constant, dV_r/dt is equal zero. Therefore, we have the following equation describing the activity of a cell.

$$\frac{d}{dt}\Delta V = \frac{g_e}{C_m}(V_e + V_m) - \frac{g_r}{C_m}(\Delta V) - \frac{g_s}{C_m}(\Delta V) \tag{4.6}$$

Each cell may then be represented by an electrically independent circuit, as in *Figure* 4.3, with x_{ik} representing the deviation of the membrane voltage from the resting potential of the cell $C(i,k)$ at the $(i,k)^{th}$ position of the lattice, and $f(x_{ik})$ denoting its firing rate. We assume that the inhibitory synapses of a cell are controlled by the activity of the neighbouring cells. We also assume that the shunting conductance of a cell is the sum of the conductance of all the individual inhibitory cells. If each one of these is proportional to the firing rate of the cell controlling it, then we can write the shunting conductance g_s of $C(i,k)$ as:

$$\frac{g_s}{C_m} = \sum_{C(k,l)\in N_r} w_{ik}^{jl} f(x_{jl}) \tag{4.7}$$

where the coefficients w_{ik}^{jl} and C_m are positive constants, $f(x_{jl})$ is the output of cell jl, and w_{ik}^{jl} is the weighting given to its inhibitory effects on cell ik. In other words, the inhibition exerted on a cell is a weighted sum of the outputs of surrounding cells within the appropriate neighbourhood. The conductance g_e is controlled by the excitatory inputs that work to increase the membrane conductance to sodium Na^+ ions. We assume that the cell input, I_{ik}, controls the excitatory current i_e, in which case we have

$$\frac{i_e}{C_m} = \frac{g_e}{C_m}(V_e + V_m) = I_{ik}(t) \tag{4.8}$$

The remaining term is the decay factor of excitation:

$$\frac{g_r}{C_m} = a_{ik} \tag{4.9}$$

Then Equation (4.6) becomes

$$\frac{dx_{ik}}{dt} = I_{ik}(t) - a_{ik}x_{ik} - \sum_{C(j,l)\in N_r} w_{ik}^{jl} f(x_{jl})x_{ik} \tag{4.10}$$

where x_{ik} represents the input intensity of cell ik, I_{ik} is its input, a_{ik} determines the excitation decay rate, w_{ik}^{jl} is the connection weight from cell jl to cell ik, and $f(x_{jl})$ is the output of cell jl.

It is clear from Equation (4.10) that the interaction comes from all cells in the local r-neighbourhood of cell (i,k), hence the local nature of information exchange. Also note that this is particular implementation of the general CNN, given by Equation (3.2).

4.4.1. A Stability of SICNN

A dynamical system is one where the state of that system changes with time and depends upon both the state itself and the system input (*Sandefur, 1990*). A dynamical system is bounded-input bounded-output (BIBO) stable if the system output, regardless of the initial state of the system, is bounded when even the input is bounded (*Ogata, 1987*). Bouzerdoum and Pinter (1993) proved that if there is symmetry in the SICNN interaction weights; i.e. $w_{ik}^{jl} = w_{jl}^{ik}$; and the activation function f is continuous, non-negative and decrescent on the entire real axis, i.e $\zeta f(\zeta) \geq 0$ for all $\zeta \in (-\infty, \infty)$; then the SICNN is a BIBO stable dynamical system.

Given a dynamical system $x = f(x,t)$ where x is the state vector, then the equilibrium state is the state x_e where $f(x_e, t) = 0$. Such a dynamical system is said to be *convergent* if every trajectory converges in the steady-state to an equilibrium point (*Sandefur, 1990*). Furthermore, if the input pattern has the same polarity (positive or negative), then each of trajectory of a SICNN converges to an isolated equilibrium point (*Bouzerdoum and Pinter, 1993*).

4.5 Classifications of SICNN

In this section we present the feed-forward and recurrent SICNNs (*Pontecorvo, 1998, pp. 60-68*), of which the latter will form the basis of our contrast enhancer. The properties of such networks are analysed. We begin by defining the feed-forward SICNN and show how it performs shunting inhibition on step edges. The effects of asymmetrical and symmetrical weights on the SICNN output are also investigated. We then proceed by showing that the steady-state of the recurrent SICNN can be found using an iterative algorithm. Finally, the recurrent SICNN's response to constant and step edge inputs are investigated.

4.5.1 The Feed Forward SICNN

The feed-forward network architecture can be represented pictorially as in *Figure* 4.4, where for simplicity only three nodes are shown. The inputs to each node include the inputs to its two nearest neighbours. In a feed-forward SICNN, the state of each cell depends only on the input signal. As we shall see, the computational steps in calculating the feed-forward SICNN output consist of a small number of matrix operations.

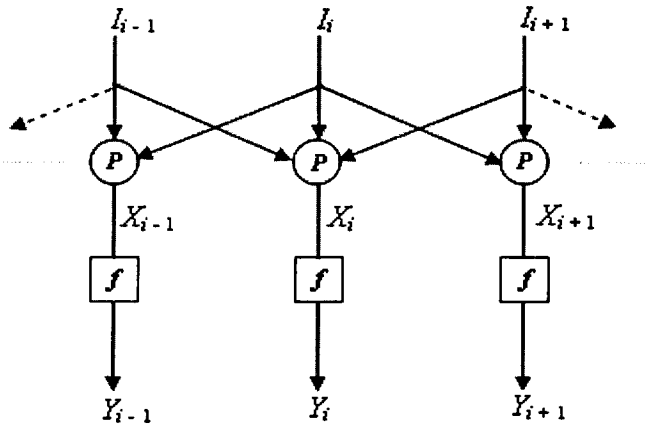


Figure 4.4 *The feed-forward SICNN architecture with node 'P' as the processing unit implementing Equation (4.12) (Pontecorvo, 1998, pp. 66).*

We previously discussed the derivation and stability of the Shunting Inhibitory Cellular Neural Network in Section 4.4. By analogy to the SICNN system given in Equation (4.10), feed-forward system can be written as:

$$\frac{dx_i}{dt} = I_i - a_i x_i - \sum_{j \in N_r(i)} w_{ij} f(I_{i-j}) x_i \qquad i = 1,2,...,M \qquad (4.11)$$

where x_i is the state of cell i, I_i is its input, a_i its decay factor f is the activation function, w_{ij} is the interaction weight between cells j and i, N_r is the neighbourhood function, and M is the total number of nodes which equals to the number of inputs.

By converting the neighbourhood function into the distance of separation between cells, or input units, the steady-state solution of Equation (4.11) can be given as:

$$x_i = \frac{I_i}{a_i + \sum_{j=-r}^{r} w_j f(I_{i-j})} \qquad (4.12)$$

where r is the neighbourhood size, which represents the range of cells that can have a direct inhibitory effect on cell i. Thus, the inhibitory effects of neighbouring cells on cell i is a weighted linear combination of their respective inputs. We should recall that, in general, there is no feedback from any cell to itself, hence $w_0 = 0$ in all cases.

Assuming that the weight distribution is space-invariant, i.e., the strengths of connections of a cell to its neighbours is dependent only upon the neighbours' relative position and not their absolute position with respect to the node of interest, then the weights can be represented by a vector $\boldsymbol{w}$, often referred to as a *template* or *mask*. The interaction between all cells can then be achieved by convolving the weight template with a function $f(I)$ of the input states I.

If Y is the output vector, X a vector of state values, f is the activation function, I a vector of inputs, w the weight template, and A the vector of decay factors, then X_i, the i^{th} element of X, is given by:

$$X_i = \frac{I_i}{A_i + [w * f(I)]_i} \qquad (4.13)$$

$$Y_i = f(X_i) \qquad (4.14)$$

where * denotes 1-D convolution, and $f(I)$ is a vector with $[f(I)]_i = f(I_i)$. Thus, the output of the feed-forward SICNN is obtained by using a few simple vector operations. These

operations are not computationally intensive, and can be implemented rapidly in either hardware or software.

4.5.1. A Step-Edge Response

Consider a step-edge input as shown in *Figure* 4.5. The shunting inhibition on this edge can be easily explained by feeding it to a SICNN system with a neighbourhood size of r=1. The state of each cell or node is inhibited by the inputs of both its nearest neighbours. The feed-forward SICNN output is then given by:

$$x_i = \frac{I_i}{a_i + \left[w_{-1} f(I_{i-1}) + w_1 f(I_{i+1})\right]} \tag{4.15}$$

where the term $[w_{-1} f(I_{i-1}) + w_1 f(I_{i+1})]$ represents the inhibitory effect from neighbouring inputs. As shown, the output of each node also depends on its input and the decay factor.

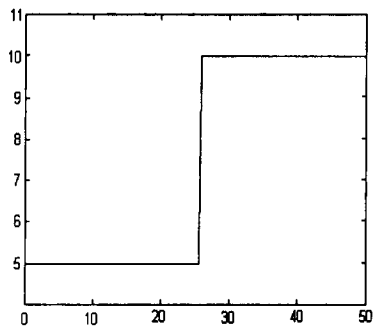
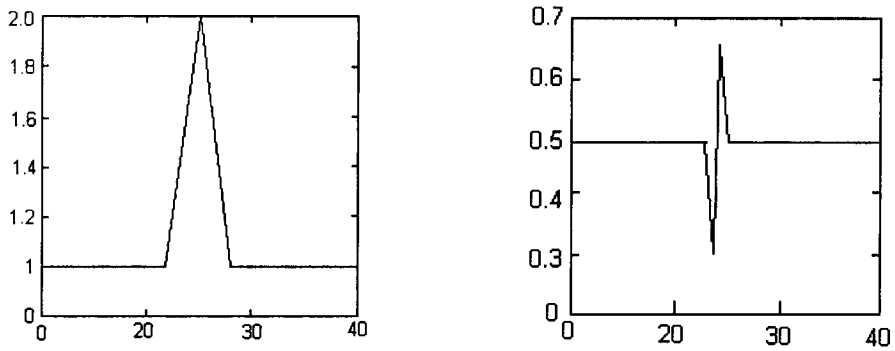


Figure 4.5 *Step edge input.*

The outputs of this feed-forward SICNN to the edge input in *Figure* 4.5 are shown in *Figure* 4.6 (*a*) and (*b*) for asymmetrical and symmetrical weights respectively.



(a) ***(b)***

Figure 4.6 *Feed-forward SICNN output (a) Asymmetrical weights (b) Symmetrical weights.*

4.5.1. B Response to Asymmetrical and Symmetrical Weights

The response properties of the SICNN to asymmetrical and symmetrical weights have been studied by *Pontecorvo (1998, pp. 68)*. We briefly describe these properties in this section.

Asymmetrical Weight Distribution

Consider first a feed-forward SICNN with an asymmetrical weight vector [0 0 1], where the inhibition comes from only the immediate left node of any given node. *Figure* 4.7 demonstrates the inhibitory effects of some nodes in different regions of a step edge. *Figure* 4.7(*a*) shows the inhibitory effects away from the discontinuity. For the nodes to the left of the discontinuity, both the input intensity and the inhibition signal are weak; hence the total output tends to be small. For nodes to the right of the discontinuity, both the input intensity and the inhibition signal are large, the total output again tends to be small. *Figure* 4.7(*b*) shows what happens at the edge point. Although the inhibition is weak, the intensity of the input is large, hence the output is large. This results in a peak in the overall output of the SICNN, as seen in *Figure* 4.7(*c*).

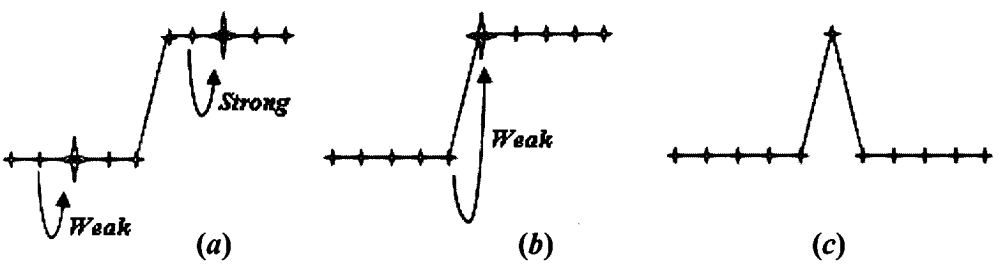


Figure 4.7 *The inhibitory effects on a step edge input for a SICNN with asymmetric weights as [0 0 1] and (c) gives the overall SICNN output.*

Symmetrical Weight Distribution

Now consider the SICNN output with a symmetrical weight vector of [1 0 1] shown in *Figure* 4.8. The effect on an input step edge away from the actual edge is shown in *Figure* 4.8(*a*). Consider first the input to the left of the discontinuity. On this part of the edge the inhibition and the intensity on each node is small, hence x_i will be small. Now consider the effects on the input to the right of the edge and away from the discontinuity.

Both the inhibitory effects and intensity are now larger, so the response here also tends to be small but may be somewhat larger than the response to the left of the discontinuity.

Now consider the case shown in *Figure* 4.8(*b*), where the node has a small amount of inhibition from the node to its left (to the left of the edge), but a large inhibitory effect from the adjacent node to the right on the edge. As the intensity of the pixel itself is still small, the net effect of the increased inhibition is to reduce the output compared to those nodes further to the left of the edge.

In *Figure* 4.8(*c*) the node again has both a large and a small inhibitory influence from its neighbouring cells, but the value of input intensity is now large, hence the output suddenly increases greatly compared to that of *Figure* 4.8(*b*). The overall output is shown in *Figure* 4.8(*d*).

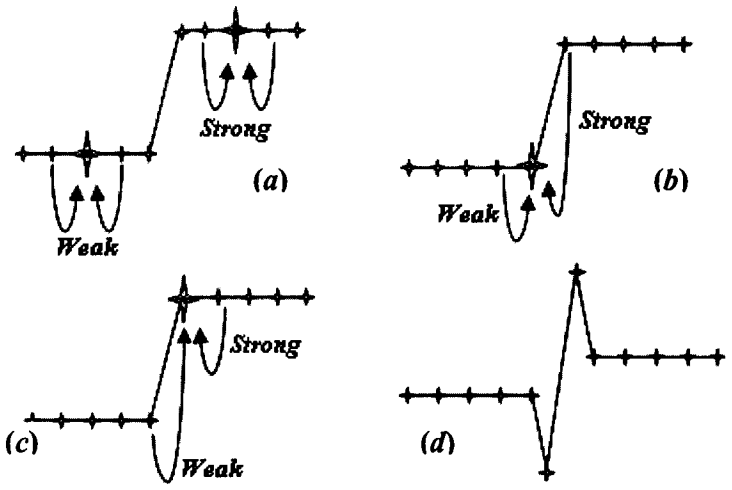


Figure 4.8 *SICNN with symmetrical weights*: (*a*), (*b*), (*c*) *show the inhibitory effects and* (*d*) *is the overall output.*

From the above discussion, the SICNN's performance is very suitable for edge detection. To detect the position of the edges, simple thresholding or zero-crossing detection can be used on the network output. Note that the number of output pixels affected by the step edge is *r* pixels to the left of the discontinuity, and *r* pixels to the right of it (or *r*-1 if we exclude the edge pixel itself).

4.5.2 The Recurrent SICNN

The SICNN derivation and stability issues were seen in Section 4.4.1. From the 1-D differential equation for the state of the cell:

$$\frac{dx_i}{dt} = I_i - a_i x_i - \sum_{j \in N_r(i)} w_{ij} f(x_j) x_i \qquad \text{i=1, 2,..., M} \qquad (4.16)$$

where x_i is the state of cell i, I_i is its input, a_i its passive decay factor of excitation, f is the activation function, w_{ij} is the interaction weight between cells j and i, N_r is the neighbourhood function, and M is the total number of nodes which corresponds to the total length of the input. Such a network is *recurrent*, as each x_i depends upon the value of the neighbouring x_j, which in turn depends upon that cell's state itself. The network can be represented pictorially as in *Figure 4.9*, where for simplicity only three nodes are shown with each node interacting directly with its two nearest neighbours.

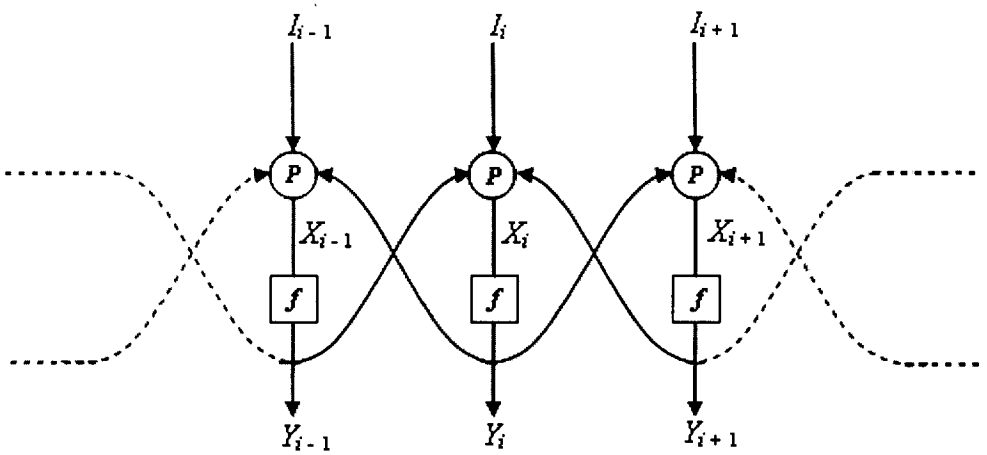


Figure 4.9 *The recurrent SICNN architecture with node 'P' as the processing unit implementing Equation (4.16). (Pontecorvo, 1998, pp. 61)*

The steady-state solution of Equation (4.16) to a time invariant input I_i satisfies

$$x_i = \frac{I_i}{a_i + \sum_{j \in N_r(i)} w_{ij} f(x_j)} \qquad (4.17)$$

Converting the neighbourhood function into the distance of separation between cells, or input units, we can write the previous equation as:

$$x_i = \frac{I_i}{a_i + \sum_{j=-r}^{r} w_j f(x_{i+j})} \qquad i = 1,2,...,M \qquad (4.18)$$

where r is the neighbourhood size. Thus, the inhibitory effects of neighbouring cells on cell i is a weighted linear combination of their respective outputs. We should recall that, in general, there is no feedback from any cell to itself, hence $w_0 = 0$.

4.5.2. A Steady-State Response

Unlike the feed-forward SICNN, the cell state in a recurrent SICCN is nonlinearly dependant upon the states of neighbouring cells. The steady-state value of each cell of the recurrent SICNN can be obtained by numerically solving the system of differential equations given in Equation (4.16). Alternatively, we can define an equivalent discrete-time dynamical system that has a steady-state solution equal to the steady-state solution of Equation (4.18). The discrete-time solution of the recurrent SICNN after the first iteration of its steady-state solution is also equivalent to the steady state output of the feed-forward SICNN. For the recurrent SICNN such a discrete-time dynamical system is described by

$$x_i(k+1) = \frac{I_i}{a_i + \sum_{j=-r}^{r} w_j f(x_{i+j}(k))} \qquad k = 1,2,...,\infty \qquad (4.19)$$

where k is the *discrete-time step*, or iteration number. The sequence is solved iteratively, i.e., given an initial estimate of the steady-state solution $x_i(0)$, we use Equation (4.19) to derive $x_i(1)$, which in turn is used to obtain $x_i(2)$, and so-on. A simple initial value to choose is $x_i(0) = I_i$, for all i. That is, the input to each node also serves as the initial value of the steady-state value for that node, provided the SICNN converges. The convergence properties of such a network will be discussed next.

4.5.2. B Convergence

It was showed by Bouzerdoum and Pinter (1993), and noted in Section 4.4.1.A, that under certain conditions the continuous SICNN converges to an equilibrium point (from a possible set of many). If the discrete SICNN given by Equation (4.19) converges, then it will converge to an equilibrium point of the continuous SICNN. Due to the nonlinear nature of Equation (4.19), the general conditions required for convergence are unknown.

However, the convergence properties can be demonstrated for specific cases; as we now see.

We begin by explaining the process of calculation and then look at some practical outputs demonstrating convergence. Consider an asymmetrical weight distribution case such as w= [1 0 0]. Using these weights, the recurrent SICNN output for a particular iteration is given by:

$$x_i(k) = \frac{I_i}{a_i + w_{-1} f(x_{i-1}(k-1)) + w_1 f(x_{i+1}(k-1))} \qquad \text{for} \quad i = 1, 2, \ldots, M$$

Now by considering the left-most node of the network, we note that because there is no node to its left, the first iteration output $x_1(1)$ only depends upon the input intensity and the decay factor of that node. As both a_1 and I_1 are constant, $x_1(k+1)$ is given as:

$$x_1(k+1) = \frac{I_1}{a_1}$$

where x_1 is constant for $k \geq 1$.

Looking at the next node, we see that $x_2(1)$ depends upon $x_1(0)$. In a general notation,

$$x_2(k+1) = \frac{I_2}{a_2 + x_1(k)}$$

Since a_2, I_2 and $x_1(k)$ are all constant, $x_2(k)$ also remains constant for $k \geq 2$.

Similarly, by induction,

$$x_3(k+1) = \frac{I_3}{a_3}$$

where x_3 is constant for $k \geq 1$.

Thus, the network converges. This same result holds for asymmetrical weights of any size, and even in the reverse order such as w = [0 0 1]. Thus, a SICNN with any asymmetrical weight distribution is convergent. Similar observation can be made even in the symmetric weight distribution case. We now look at some practical outputs suggesting the convergence property of the SICNN.

Below demonstrated are two experiments conducted to show convergence property of discrete-time SICNNs. The horizontal axis of the figures represents the number of iterations and the vertical axis gives the value of x_i at different instances. *Figure* 4.10(*a*) shows the time evolution of the response of a SICNN with symmetric weight distribution for a constant input equal to 10. It is clear from this figure that the system converges at about 21 iterations. *Figure* 4.10(*b*) shows the time evolution of the response of a discrete-time SICNN with asymmetric weights for a constant input equal to 10. Convergence to the steady-state response is at about the 16^{th} iteration.

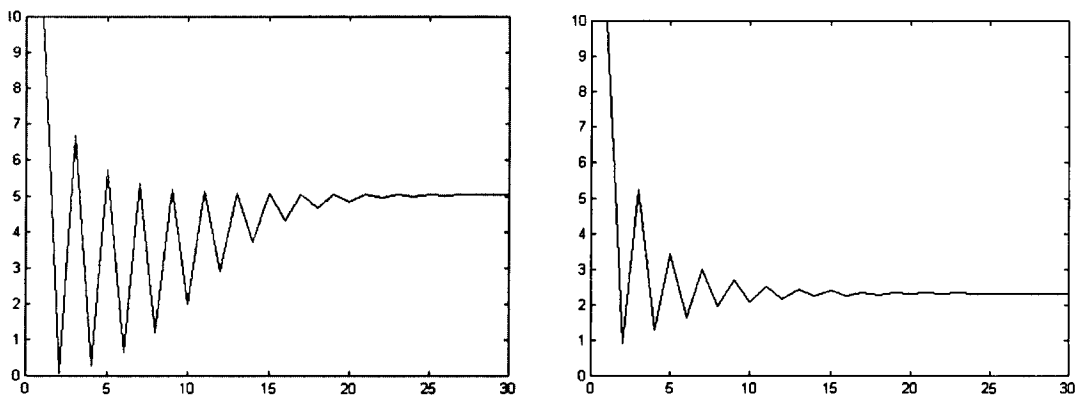


Figure 4.10 *Values of x_i (k) of Equation* (4.3) *until convergence,*

(*a*) *for an input of intensity* 10 *with* w = (1 1 0 1 1).

(*b*) *for an input of intensity* 10 *with* w = (0 0 1).

The neighbourhood sizes used are 2 and 1 for *Figure* 4.10(*a*) *and* (*b*) with weight distributions to be w = (1 1 0 1 1) and w = (0 0 1) respectively. This shows clearly that the neighbourhood size and type of weights used play a big role in stabilizing the system.

4.5.2. C Step Edge Response

In the earlier sections, we have seen the response of a feed-forward network to step edges. *Pontecorvo* (*1998, pp. 64*) demonstrated the step edge response properties of a recurrent SICNN and analysed its similarities with the feed-forward system. We begin the

review of this analysis by using a numerical example quoted to generate the appropriate responses.

Consider the case of an input sequence of length $2M$, with the step discontinuity occurring exactly half-way along its length. Then, the first M nodes of the SICNN have a lower step intensity I_L as their input, and the remaining M nodes have the upper step intensity I_U as their input. Thus,

$$I_i = \begin{cases} I_L & if \quad 1 \le i < M \\ \\ I_U & if \quad M+1 \le i \le 2M \end{cases}$$

However, treating the step-edge as a sectioned constant would enable us to temporarily ignore the step discontinuity. By neglecting the boundary effects of the sequence also, the output of the system can be computed in an analogous manner. That is, the output of the nodes in each of the constant sections of the step input can be found using the iterative method outlined previously, with each node's state approaching its steady-state value as the number of iterations increases. Obviously, the interesting effects occur at the edge discontinuity.

Recalling from Equation (4.19), the SICNN output of node i for the k^{th} iteration and a neighbourhood $r = 1$ is given by:

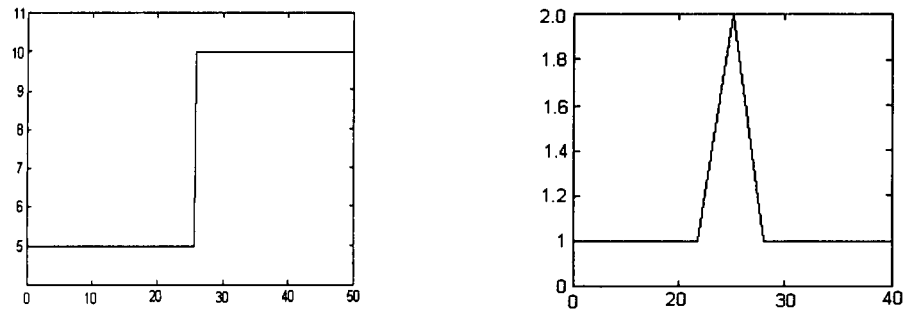
$$x_i(k) = \frac{I_i}{a_i + w_{-1} f(x_{i-1}(k-1)) + w_1 f(x_{i+1}(k-1))} \qquad for \quad i = 1,2,\ldots,M$$

As a numerical example, if we choose $I_L = 5$, $I_U = 10$, $w = [0\ 0\ 1]$, $a_i = 0.1$ for all i, f a linear function, and $M = 25$, then the edge point response of a step input shown in *Figure*4.11(a) is given by:

$$x_{26}(1) = \frac{I_{26}}{a_{26} + \left[(1 \times I_{24}) + (0 \times I_{27})\right]} = \frac{10}{0.1 + 5} = 1.96$$

Similarly, the outputs of neighbouring nodes are $x_{25}(1) = 0.980$ and $x_{27}(1) = 0.990$.

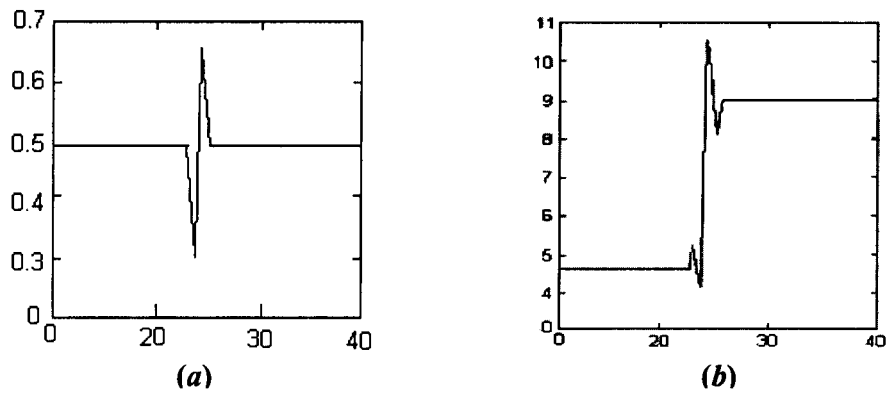
Figure 4.11 shows the initial step input and the first iteration output for the SICNN with an asymmetrical weight distribution.



(a) *(b)*

Figure 4.11 *(a) Step edge input; (b) The first iteration recurrent SICNN output with asymmetrical weights.*

Figure 4.12 shows the first and second iteration outputs for the SICNN with symmetrical weight distribution e.g., w = [1 0 1]. By comparing with *Figure* 4.6, it can be said that the first iteration outputs of the recurrent SICNN to edge inputs are identical to the corresponding feed-forward SICNN outputs for asymmetrical and symmetrical weights, respectively.



(a) *(b)*

Figure 4.12 *The recurrent SICNN output after (a) one iteration and (b) two iterations of Equation* (4.19).

Clearly the output after the first iteration of the SICNN with asymmetrical weights, as shown in *Figure* 4.11(*b*), can be used for edge detection if the maximum output is located. For the output of the SICNN with symmetrical weights as shown in

*Figure*4.12(*a*), the first iteration output can be used for edge detection by finding the 'zero- crossings' in the output, while the second iteration output as seen in *Figure* 4.12(*b*) is more suited for image enhancement, in particular edge enhancement. *Figure* 4.13 shows a similar application of the SICNN on a two dimensional image. The first two iteration results are shown to demonstrate the edge detection and enhancement capabilities of the SICNN.

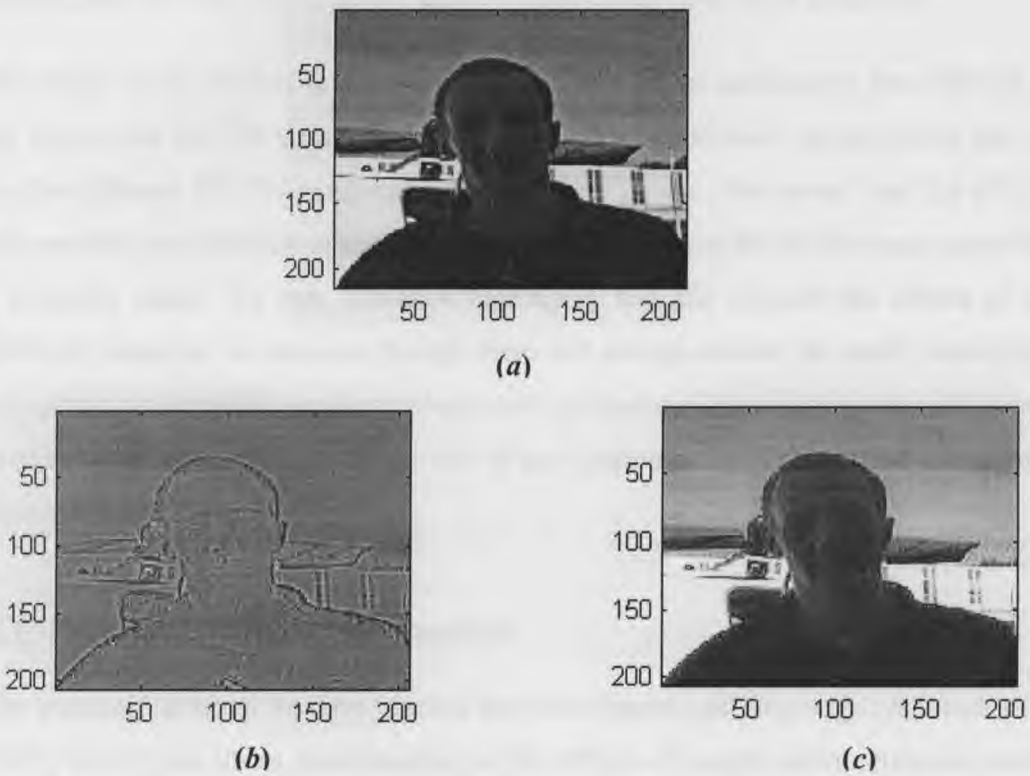


Figure 4.13 *Two dimensional implementation of SICNN.* (*a*)*Original image* (*b*) *First iteration output showing Edge Detection* (*c*) *Second iteration output showing Enhancement.*

From observation, it can be generally said that the odd iterations are appropriate for detecting edges while the even iterations are more adaptable to enhance edges. The adaptability of the SICNN is better demonstrated while discussing the dynamic compression properties in the next section.

4.6 Advantages of SICNN

The SICNN has many tuneable parameters, such as the weight distribution, decay factor of excitation, activation function, and neighbourhood size. This gives us greater flexibility in adapting the network to a particular type of input edge and/or noise, hence achieving better results than most systems that cannot adapt or have very few tuneable parameters. The SICNN's parameters can be tuned to such an extent that the network could be used for many applications such as enhancement and edge detection.

A drawback of the SICNN is that its output intensity varies nonlinearly for different step edge inputs and SICNN parameters. Thus, its edge enhancement performance will also vary for different SICNN parameters and different inputs. We cannot test the effect of every possible combination of these parameters and inputs on the performance since there are infinitely many. We can, however, thoroughly test and observe the effects of each individual parameter in isolation, though there will always remain the small possibility of overlooking a potentially good combination of parameters. Fortunately, this will become less of an issue as we understand the role of each parameter on the contrast enhancement performance.

4.6.1 Dynamic Range Compression

In the previous sections we have studied the convergence and step edge responses of the SICNN, which gave us an understanding of the effects of weight distribution and number of iterations on the SICNN output. We now investigate the dynamic range compression property of the SICNN (*Cheung et al., 1999*) which would give us an overall idea of the superiority of the SICNN as compared to more traditional enhancers like the *log* transformation system.

To understand this, a ramp, with a slope of one, is used as input to the SICNN. All cells used here are assumed to have a constant decay factor and the sum of the weights is made equal to one. In *Figure* 4.14, Graph (*d*) shows the identity transformation of the ramp input; Graph (*a*) gives the first iteration SICNN output; Graph (*c*) gives the steady-state

output of the SICNN with a neighbourhood of five cells; and Graph (*b*) is the output of a *log* transformation system.

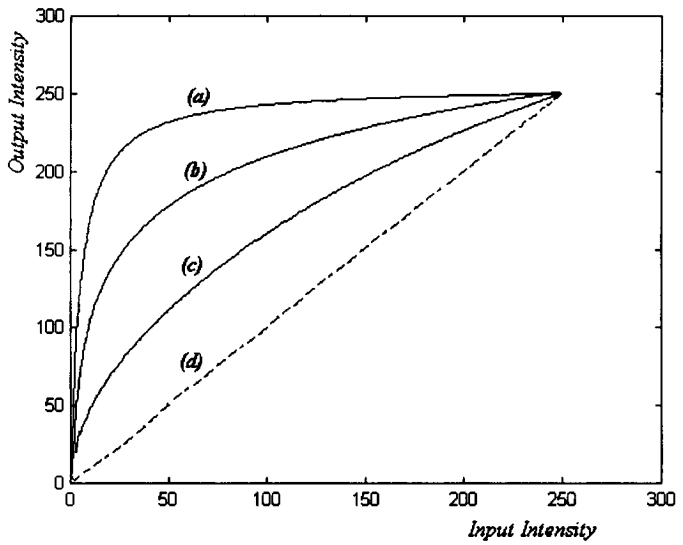


Figure 4.14 *Dynamic compression of SICNN.*

From this figure, it can be suggested that the first iteration output of the SICNN is most suited to edge detection, while the results of the other iterations can be used for edge and contrast enhancement. Although not shown here, the SICNN can achieve a range of transformations between graphs (*a*) and (*c*), depending on the network parameters such as the number of iterations and its neighbourhood size. Furthermore, the decay factor and interaction weights play an important role in reshaping these curves, which influences the intensity of enhancement.

4.7 Conclusion

The cellular structure of CNNs and their local interactions make them particularly suitable for modelling visual phenomenon such as Mach bands, as each cell of these CNNs consist of the equivalent electrical circuit of a biological neuron.

We began this chapter by understanding the concepts of linear and non-linear lateral inhibition. From the insights of a biological neuron, we saw the derivation of an electrical circuit having equal interaction capabilities. The cell state equation derived using this

circuit was reviewed. This state equation clearly showed inhibition of the shunting or multiplicative type. The stability of such a model was also discussed.

Both the feed-forward and recurrent SICNNs were studied in this chapter. We defined the feed-forward SICNN, and illustrated the shunting inhibitory nature of the SICNN on step edge signals. We also explained its properties using symmetrical and asymmetrical weights.

We then studied the recurrent SICNN, and showed that its steady-state solution is nonlinear, i.e. the output of any cell is dependant on the output of its neighbouring cells. Instead of using computationally intensive calculations to solve such a system, the use of an iterative method to reach the steady state was demonstrated. The convergence properties of such a system using symmetric and asymmetric weights were graphically explained.

The recurrent SICNN's response to step-edges after one iteration was found to be very similar to that of the feed-forward SICNN. It was even iteration outputs of the recurrent SICNN that interested us the most. Upon analysis in both 1-D and 2-D, it was concluded that the odd iteration outputs of the recurrent SICNN were suitable for edge detection while the even iteration properties showed enhancing properties. For this reason, the recurrent SICNN was chosen as the foundation for the enhancement application to be performed in this thesis. Finally, we demonstrated the advantages and superiority of such a system over other traditional techniques by comparing their dynamic range compression capabilities.

Chapter 5

SICNN Design for Ultrasound Imaging

5.1 Introduction

When a real image is acquired by a camera of some sort, there is invariably some degradation in the image quality. This may be due to bad lighting, poor camera calibration, or low dynamic range in the camera. Sometimes even the techniques used to acquire these images also could contribute to the degradation of information in the image. Targeting the exact source of degradation would drastically reduce the burden of improving the quality of the image.

In this chapter, we begin by examining the ultrasound technology in Section 5.2. Its operating principles and applications in medical therapy and diagnoses are reviewed. We focus on the use of ultrasound scanners in detecting the various tissues and organs in a human body. The detected ultrasound patterns are categorised and the ambiguities in the detection process are highlighted.

In Section 5.3, we proceed to the goal of this study, i.e. designing the SICNN to overcome the ambiguities in the speckle patterns of the scanned ultrasound images. We begin by simplifying the steady-state solution of the digital SICNN where the various parameters affecting the SICNN's performance are highlighted. Targeting the problems

in ultrasound images, we design a dynamically varying parameter system to minimize the speckle ambiguities without significantly disturbing the true content of the images.

5.2 Ultrasound Imaging

In medicine, ultrasound is used for diagnosis and therapy, in which very high frequency sound is directed into the body. Though the therapeutic ultrasound technique has limited applications, the diagnostic applications are popularly used in science. In this section, we first describe the operation principles of Ultrasound Imaging (UI) and its applications. Then, the problems in UI are highlighted by explaining its detection theory.

5.2.1 Operation Principles

Ultrasound is produced by a rapidly oscillating crystal and, with a frequency greater than 20 kHz, is inaudible to humans. The crystal vibrates about 500 times per second with each vibration lasting for about one millionth of a second (*Lewis, 1999*). A transducer is used to transmit the sound and receive the echoes. The transducer is held in close contact with the skin, on which a jelly-like substance is smeared in order to improve the acoustics. The soft-tissue surfaces reflect the sound and the resulting patterns of sound reflection are processed by a computer to produce an image on a screen or on photographic film.

5.2.2 Diagnostic Applications

The UI technique is of no use in determining conditions of the bones or lungs as air, bone, and other calcified tissues absorb nearly all the ultrasound beam. However, UI is a useful technique for diagnosing cysts, bladder structures, biliary systems, and other fluid filled structures, as the ultrasound beams are well conducted in fluids. The best-known application of ultrasound is the examination of the foetus in the amniotic sac during pregnancy. Unlike X-rays, ultrasound is safe during pregnancy; it poses no risk to either mother or baby.

Ultrasound can be used to examine various body organs, including the arterial system, the heart, the pancreas, the peritoneal cavity, the urinary tract, the ovaries, the venous system, the brain, and the spinal cord. When ultrasound is used to examine the heart, the technique is known as echocardiography. Echocardiography is used to study congenital heart disease, coronary artery disease, tumours of the heart, and other cardiac disorders. Ultrasound can also be used to guide surgical procedures, for example during amniocentesis or the insertion of a biopsy needle into a particular area. Doppler ultrasound is used to measure the flow of body liquids, for example the blood flow.

5.2.3 Ultrasound Therapy Applications

Ultrasound is also used in the treatment of various disorders of deep tissue. It can break up gallstones or stones in, for example, the kidney. Ultrasound is also useful when used in conjunction with physiotherapy to treat soft-tissue damage and other injuries.

5.2.4 Speckle Patterns

In non-invasive UI technique, the advantage of using ultrasound pulses is also a disadvantage. The interaction of the ultrasound pulses with different tissue types, gives rise to various interference phenomena.

Speckle patterns that simulate the structure of tissue originate from constructive interferences. The destructive interferences that bear little resemblance to the actual acoustical tissue microstructure are known as speckle noise. The speckle causes the degradation of the contrast resolution (*Mc Dicken, 1991*).

In ultrasound imaging, contrast resolution can be explained as the minimum change in reflectivity of the tissue, which can be depicted in an image (*Macovski, 1983*). Tissue deformation such as lesion may provide only a low contrast with the adjacent host organs. Increasing the contrast resolution in such a case would mean to lower the speckle noise from the relevant speckle patterns.

5.2.5 Problems in Ultrasound Imaging

We have seen from the working principle of the ultra sound, that air, bone and other calcified tissues absorb nearly all the ultrasound beam. It is only the fluidic structures of the body that are highly conductive. Damage to these conductive tissues would result in a heavy loss in their reflective property.

Consider a scanned ultrasound image as shown in *Figure* 5.1. This figure shows two quadrants of a heart separated by a fine gap in between. This gap has been highlighted for clearer understanding. The delivered speckle patterns can be classified into three categories:

- The first category patterns deal with even reflections. Here we only consider the quadrants of the heart whose structural properties are similar. As the ultrasound reflected by each of these quadrants is almost constant, the speckle patterns shown are dark in nature, showing that there is no unevenness in the organ encountered.

- In the second category we deal with uneven reflections. Each quadrant of the heart is separated by a minute gap filled with conducting fluids. When an ultrasound pulse encounters such discontinuities in the heart, it gets reflected back with some unevenness. These uneven pulses are displayed as white patches in the plotted image.

  Tissue deformation also causes a loss in the reflectivity property. Pattern changes similar to the second category will also be observed during such abnormalities. Differentiating such disorders from the regular discontinuities of organs is usually left to the radiologists.

  Thus, the results from both the above categories form relevant information to the radiologist.

Figure 5.1 *Ultrasound image showing two sections of the heart. The highlighted portions show the quadrants of the heart (CDR Hospitals, 2003, India).*

- There is a third category in ultrasound images. Upon analysis, it is the information in this category that is undesired. This irrelevant information is caused due to the interaction of a particular ultrasound pulse with various layers of tissues and fluids. Though all the tissues and fluids are conductive in nature, the reflected ultrasound wave gets partially degraded because of penetration through various layers.

Visually, these waves are presented with partial intensity. In other words, the cells representing these waves are grayish in nature and normally lie around the range of 90-170 (scale of [0, 256]) instead of the extreme dark or the extreme bright. This is biologically considered as noise. As this unwanted information forms part of the speckle patterns obtained and is similar in nature; it is also termed as speckle noise.

Though speckle noise is the main problem in ultrasound imaging, there are other problems such as poor dynamic range and excessive analog darkening/brightening, which lower the contrast of the UI image. All these problems are addressed using various enhancement schemes.

5.3 SICNN Design for UI Enhancement

Degradation of the quality of an image is almost inevitable in every scanner or transmitter. This brings out the need to use digital enhancers of various levels to regain the desired quality of the image. Enhancement does not always have to make the image visually more appealing. In our case, enhancing ultrasound images is more towards tuning the image to highlight clearly all the discrepancies recorded by the ultrasound scanner.

We have just seen the complexities in forming the ultrasound images. Solving such complex problems would require many parameters to combine appropriately to generate the right proportions of the outputs. As seen in the earlier chapters, the SICNN has many adjustable parameters. This is one of the main strengths of the network. The number of combinations that can be produced by these parameters makes the SICNN flexible enough to adapt itself to many applications, contrast enhancement being one of them.

The SICNN has to be designed to perform intense simulations and output a visually and quantitatively suitable enhanced contrast pattern. From the response properties of the SICNN we have seen that by varying the various parameters, we can obtain different outputs.

In this section, we first re-state the steady-state solution of the recurrent SICNN and summarize the iterative method to compute the network output. The steady-state solution of the digital SICNN is then simplified, where the different parameters affecting the SICNN's performance are highlighted. Finally the values of these parameters are selected to achieve the desired output for contrast enhancement.

5.3.1 Digital SICNN

The SICNN for enhancement used in this thesis is recurrent in nature. The recurrent SICNN is non-linearly dependant upon the outputs of the neighbouring cells:

$$x_i = \frac{I_i}{a_i + \sum_{j=-r}^{r} w_j f(x_{i+j})} \qquad i = 1,2,...M \tag{5.1}$$

To solve this equation, an iterative method is usually used. The immediate output at each iteration is computed as follows:

$$x_i(k+1) = \frac{I_i}{a_i + \sum_{j=-r}^{r} w_j f(x_{i+j}(k))} \qquad k = 1,2,...\infty \tag{5.2}$$

From *EQ* (5.2), we can see that there are several factors that can influence the performance as well as the implementation of the digital SICNN. These factors are the value of the decay factor a_i; the connection weights w, the activation function and the iteration number k. The neighbourhood function is indirectly defined by the weight matrix.

It has been proven that under appropriate conditions, this iterative computation converges to an equilibrium point (*Bouzerdoum and Pinter, 1993*). Note that the SICNN can have many equilibrium points.

Assuming that the activation function is $f(x) = x$, *EQ* (5.2) can be simplified as (*Cheung, et. al., 1999*):

$$X(k+1) = \frac{I}{A + W * X(k)} \qquad for \quad k = 1,2,...\infty \tag{5.3}$$

where $\boldsymbol{I}$ is a matrix representing the input image; $\boldsymbol{X}$ is a matrix containing the states of the cells; $\boldsymbol{A}$ is a matrix containing the decay values of the cells; $\boldsymbol{W}$ is the weight matrix representing the connection strengths and neighbourhood size; the * denotes the convolution operator; k denotes the iteration number.

To obtain an enhanced 2-D image R_i, the SICNN output is first multiplied by an input feedback factor αI and then added to the input image I, as shown in *Figure* 5.2. Here α is a scaling factor.

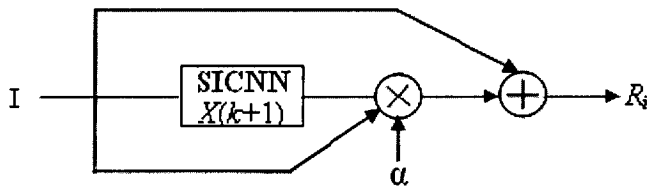


Figure 5.2 *Block Diagram to obtain two dimensional outputs.*

Thus, the 2-D SICNN enhanced response is given as

$$R_i = I(1+\alpha X(k+1) \tag{5.4}$$

The SICNN has three main factors: the decay factor, the background effects (weights and the neighbourhood size), and the iteration number. In the following sub-sections we examine the effects of each parameter on the contrast enhancement capability of the SICNN.

5.3.2 Decay Factor

Images in every application have some specific features that contribute to their resolution. Understanding the need and working of these features would help not only get a better grasp of their applicability, but also would help reduce any possible problems in their applications. In our case, we have already studied how the ultrasound imaging system works and detects abnormalities in the body. We also learnt the formation of speckle noise due to the degraded reflections of sound waves by the various tissue layers in the body.

Apart from solving the general enhancement problems in imaging, the SICNN enhancer is mainly focussed on reducing the ambiguities in the speckle patterns in the ultrasound imaging. As the possible combinations of the SICNN parameters could be enormously large, we deal with each parameter individually. Of the three listed parameters, the decay factor is used to control the global enhancement of the cells at various intensity levels.

However, assuming a condition where the decay factor is almost zero and having an average background effect as the input itself, an enhancement of 100% can be obtained. But, this is only possible for the specified ideal conditions.

Having the same rate of enhancement throughout the image would probably increase the contrast but will not be efficient enough to lower the speckle ambiguities. Moreover, during digital simulations, excessive enhancement to the sharper gray levels (nearing 0 or 256) would create an overflow of intensities, beyond the maximum range of 0-255. For these reasons, we use a dynamically varying decay factor for the different gray levels.

From the theory of the addressed problem, the gray shades of the range nearing 128 are undesirable. Relevant experimentation revealed that the approximate range of shades between 100 and 170 cause the maximum blurring. These shades need the maximum enhancement while the sharper shades need least enhancement. As the decay factor is inversely proportional to the rate of enhancement, we choose a decay function that is lowest in the middle and increases exponentially towards the two ends.

$$a_i = cp^q \qquad for\ all \quad 0 \le p \le 1 \tag{5.5}$$

$$q = \begin{cases} I_i & for \quad I_i \le t \\ 256 - I_i & otherwise \end{cases}$$

where a_i is the decay factor, c is a scaling constant and p represents the range of cells needing maximum effect, t is the threshold point marking the point of maximum enhancement.

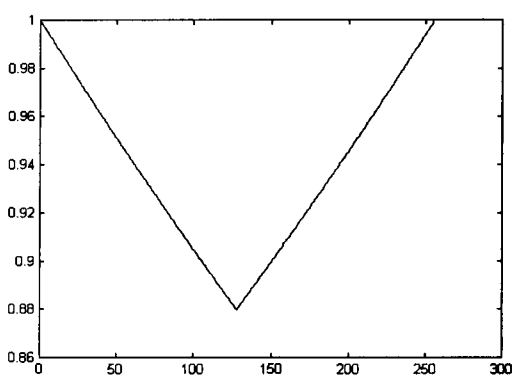
By varying the function parameters, the above decay factor expression can be tuned to obtain a wide variety of impact patterns. Now let us experiment with three cases which satisfy the approximate range of maximum enhancement. Assuming that the function parameters are constant at t = 128 and c = 1 for all three cases, the decay factors of the input I_i for different values of p are given in *Table* 5.1:

Table 5.1 *Decay factors for the different cases of x (other parameters given in text).*

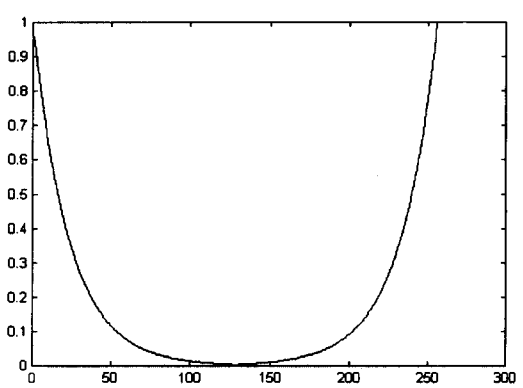
I_i	(*a*) a_i *for* x=0.999	(*b*) a_i *for* x=0.962	(*c*) a_i *for* x=0.918
1	0.999	0.962	0.918
10	0.9891	0.653	0.3902
20	0.9792	0.4433	0.1658
30	0.9695	0.3009	0.0705
40	0.9598	0.2043	0.03
50	0.9503	0.1387	0.0127
60	0.9408	0.0941	0.0054
70	0.9314	0.0639	0.0023
80	0.9222	0.0434	0.00097
90	0.913	0.0294	0.00041
100	0.9039	0.02	0.00017
110	0.8949	0.0136	0.00007
120	0.886	0.0092	0.00003
130	**0.8824**	**0.0079**	**0.00002**

Continued...

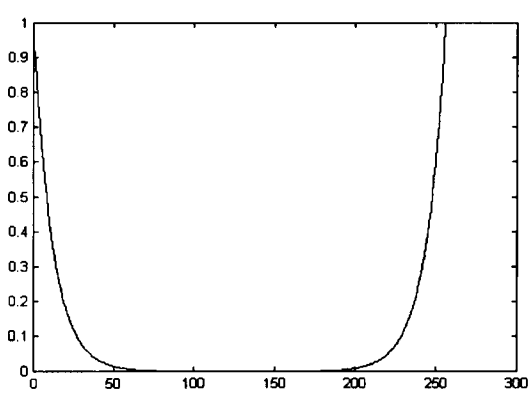
I_i	(*a*) a_i *for* x=0.999	(*b*) a_i *for* x=0.962	(*c*) a_i *for* x=0.918
140	0.8913	0.0116	0.00005
150	0.9003	0.0171	0.00012
160	0.9093	0.0252	0.00029
170	0.9185	0.0371	0.00069
180	0.9277	0.0547	0.0016
190	0.937	0.0806	0.0038
200	0.9465	0.1187	0.009
210	0.956	0.1749	0.0213
220	0.9656	0.2577	0.0501
230	0.9753	0.3796	0.1178
240	0.9851	0.5593	0.2771
250	0.995	0.8239	0.6519
255	0.999	0.962	0.918



(*a*)



(*b*)



(*c*)

Figure 5.3 *Plots of the decay factors for the corresponding cases in Table 5.1 with I_i on x-axis and a_i on y-axis.*

Figure 5.3 (*a*), (*b*) and (*c*) show the decay factor plots for x = 0.999, 0.962 and 0.918 respectively. From observation, the decay factor patterns for all cases are lowest at the given threshold (approximately 128) and highest at the two ends. However, the choice of the most appropriate pattern can only be concluded by simulating the SICNN outputs R_i for the three cases of x.

As there would be an infinitely large combination of neighbourhoods, the background effects are assumed to be the same as the input I_i for the rest of the decay factor design. The direction of enhancement is assumed to be positive. By substituting the designed decay factor values and a scaling constant of $\alpha = 0.333$ in *EQ* (5.4), the R_i values for the different cases of x are listed in *Table* 5.2.

Table 5.2 *SICNN outputs for the different cases of x (other parameters given in text).*

I_i	(*a*) R_i *for* x=0.999	(*b*) R_i *for* x=0.962	(*c*) R_i *for* x=0.918
1	1.0001	1.0001	1.0001
10	11.0129	11.0195	11.0325
20	21.0473	21.1042	21.2762
30	31.104	31.3326	32.3716
40	41.1835	41.8484	46.1624
50	51.2865	52.901	66.1242
60	61.4135	64.8958	94.0993
70	71.5652	78.4356	125.289
80	81.7421	94.3081	153.667
90	91.9449	113.34	178.216
100	102.174	136.073	200.349
110	112.431	162.365	221.291
120	122.715	191.228	241.698
130	133.016	214.338	261.785

Continued...

I_i	(*a*) a_i *for* x=0.999	(*b*) a_i *for* x=0.962	(*c*) a_i *for* x=0.918
140	143.31	220.09	281.495
150	153.62	223.709	300.818
160	163.946	225.717	319.244
170	174.287	226.818	335.648
180	184.642	227.747	347.705
190	195.012	229.129	351.344
200	205.395	231.397	341.812
210	215.791	234.771	318.764
220	226.199	239.298	291.116
230	236.619	244.905	270.53
240	247.05	251.457	261.213
250	257.493	258.799	260.776
255	261.6723	261.922	262.2443

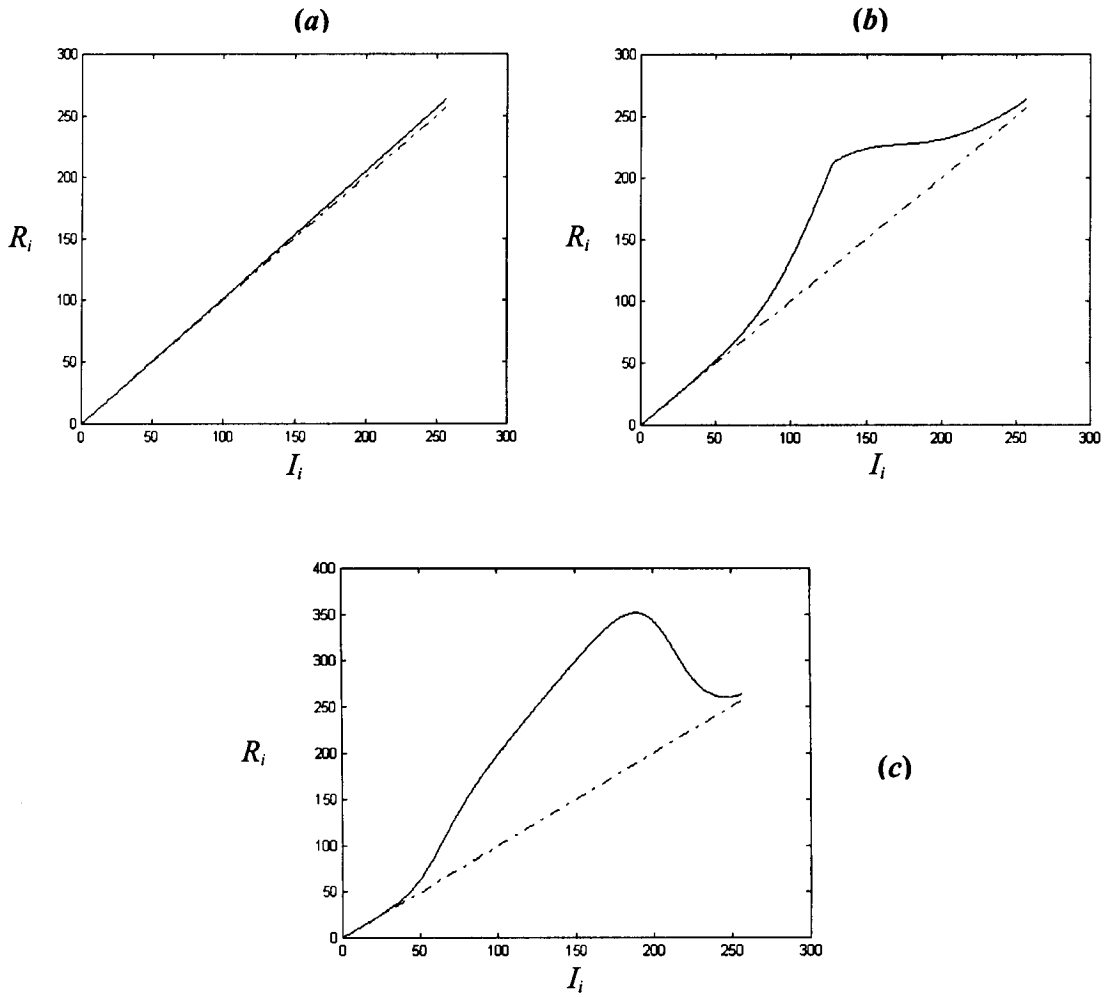


Figure 5.4 *SICNN enhanced response R_i (solid line) vs input intensity I_i for the corresponding cases in Table 5.2. Dotted line shows $f(I_i)= I_i$.*

Figure 5.4 shows the plots of R_i (solid line) for the three cases of x. A plot of I_i *vs.* I_i (dotted line) is also shown on each of the graphs to observe the variation of the output from the input.

Now let us apply a simple analogy to the obtained R_i values, based on factors such as extent of enhancement and intensity cross-overs in the outputs. The *extent of enhancement* can be explained as the difference between the output R_i and the input I_i and the *intensity cross-over* is a situation where the $R_i < R_{i-1}$. Cross-overs are not desired as they would change the flow of the intensity pattern.

- Considering case (*a*) (x=0.999), it can be observed that though there is no cross-over in the outputted intensities, the extent of enhancement is extremely low.

- In case (*c*) (x=0.918), though the extent of enhancement is very high, the R_i values of the input intensities 130-255 exceed the maximum intensity range of 0-255. In such a case, all these shades would be displayed as 255, leading to a huge loss in the contrast. Even if this can be corrected by scaling the SICNN output, there is intensity cross-overs for all inputs beyond 190, as can be observed in *Figure* 5.4(*c*).

- Case (*b*) (x=0.962) delivers more appropriate outputs by assuring no cross-overs and maintaining a reasonably high extent of enhancement. Thus, we choose the decay factor pattern in case (*b*) in future implementation.

Figure 5.5 shows a plot of I_i vs. X(k+1) for the decay pattern in case(*b*). This global enhancement pattern controls the overall required enhancement rate $X(k+1)$ and even overrides the background effects as will be demonstrated in the next section.

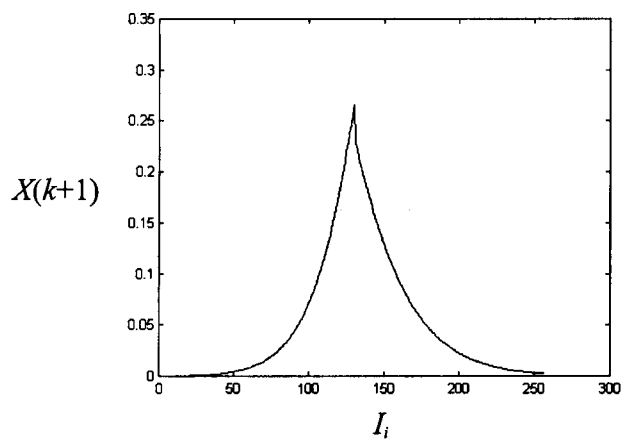


Figure 5.5 *Enhancement rate pattern for p=0.962, c=10000, t=128 and background effects to be the same as input* I_i.

5.3.3 Weights

The eccentricity of the SICNN compared to other traditional techniques is its use of background effects. We have seen that the role of the decay factor was to globally control the enhancement pattern and adapt the network to a given application. Unlike the decay factor's global design, calculating the local effects is more complex. The background effects are controlled by the weights between the cell under consideration and its neighbours. The net weight of the neighbourhood is a factor of other parameters such as the neighbourhood size and the neighbouring cell intensities.

Every neighbour has a different impact on the cell. Defining the impact weights for the thousands of individual cells is an impractical task. Thus, it is important to establish a dynamically varying weight structure. Two criteria are identified to confine such a weight structure: one of them being the perceived intensity of the neighbourhood and the other being the direction of enhancement.

We begin by computing a neighbourhood matrix of radius r. Let the pixel under consideration i be the centre of the matrix and let j be the neighbour surrounding it. Every selected neighbourhood could contain dark and bright shades. Since the system has to vary dynamically, the intensity of the cell under consideration I_i, is chosen to be the threshold for differentiating between the shades. If a particular neighbour is darker than the threshold, it is considered as a dark cell and if the neighbour is brighter, it is considered as a bright cell.

In the given neighbourhood, we compute $z_{i,j}$; the absolute values of the intensity difference between cell i and each of its neighbours. By using the difference in intensities and not the actual cell intensities, a higher enhancement rate can be ensured for the low contrast cells. This is because the background effects are inversely proportional to the enhancement rate.

As shown in *Algorithm* 5.1 the overall sum of $z_{i,j}$ for the darker cells nz and for the brighter cells pz are calculated separately. If the sum of $z_{i,j}$ for darker neighbours is higher than that of the brighter neighbours, the neighbourhood is perceived as a darker region. A bright neighbourhood is recorded in the opposite case. Based on this result, the weights

are assigned a positive or negative enhancement direction respectively. This direction is given by the variable *dir* in the algorithm below.

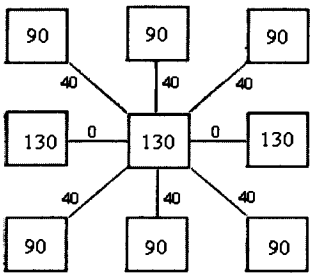
Algorithm 5.1 *Algorithm that calculates the background effects*

- Set threshold as the input (pixel under consideration, i).
- Define neighbourhood matrix (r, r).
- If neighbour z_1, is less than the threshold, then assign the difference (i-z_1) as the effective negative neighbour (nz_1).
- If neighbour z_1, is greater than the threshold, then assign the difference (z_1-i) as the effective positive neighbour (pz_1).
- Calculate the sum of the effective positive and negative neighbours separately (*pz* and *nz* respectively).
- Calculate the mean positive and negative neighbourhoods (*mpz* and *mnz*).
- Determine the direction of enhancement (*dir*). If the total negative neighbourhood (*nz*) is greater than the total positive neighbourhood (*pz*), then the direction of enhancement is positive. The direction is negative in the opposite case.
- If the direction is positive, then the mean negative neighbourhood (*mnz*) is assigned as the net background effects (*bf*) and if the direction is negative then the mean positive neighbourhood (*mpz*) is taken as the net background effects (*bf*).
- Finally, the direction (-1 or +1) is multiplied to the $X(k+1)$ value computed using the decay factor (*df*) and background effects (*bf*).

The remaining part of the background influences is to calculate the effective intensity of the neighbourhood. This is given by computing the mean n_{i-j} of the dominant neighbourhood (*mnz* for darker cells and *mpz* for brighter cells). Finally, the direction of enhancement is multiplied to the $X(k+1)$ value computed using the decay factor (*df*) and background effects (*bf*).

Example:

Figure 5.6 (*a*) shows a portion of a gray scale image with an intensity of 90 on the outer block and an intensity of 130 on the inner block. Consider a neighbourhood of size 3 x 3 as shown in *Figure 5.6* (*b*). The cell intensities are given in the boxes and the numbers on the lines joining the centre cell give the $(i - z)$ values.



(a) ***(b)***

Figure 5.6 *(a) Portion of a gray scale image with an intensity of 90 on the outer block and an intensity of 130 on the inner block. (b) Neighbourhood matrix with* r =3.

By implementing Algorithm 5.1 on these values, the total negative and positive neighbouring intensity variations (*nz* and *pz* respectively)are:

$nz = 40 + 40 + 40 + 40 + 40 + 40 = 240$

$nzc = 1+1+1+1+1+1 = 6$

Similarly, $pz = 0$ and $pzc = 2$

where *nzc* and *pzc* are the number of cells lesser or greater that the centre cell.

The mean intensity variations *mnz* and *mpz* are then given by:

$mnz = 240/6 = 40$

$mpz = 0$

Since $nz > pz$

$dir = 1$ and $bf = 40$

From the decay factor values of case (*b*), the final decay factor for I_{130} is:

$df = 79$

By substituting *bf*, *df* and I_i values into the SICNN equation, we get

$$X(k+1) = \frac{130}{79 + 40} = 1.092$$

With an α =0.333, by substituting the $X(k+1)$ value into $EQ(5.4)$,

$R_{130} = 130(1 + 0.333 \times 1.092) = 177.27$

Hence, the new iteration output would be:

90	90	90
130	177	130
90	90	90

Now by applying this algorithm to a complete two dimensional image and using the designed decay factor, the $X(k+1)$ plot for each intensity is as shown in graph (*b*) of *Figure* 5.7.

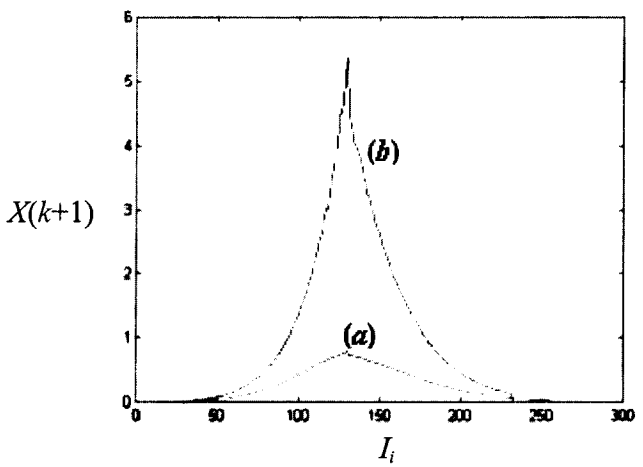


Figure 5.7 *Graph (a) shows the $X(k+1)$ values with constant background effects as I_i and graph (b) shows the $X(k+1)$ of each intensity with background effects and designed decay values for a 2-D image.*

Graph (*a*) shows the $X(k+1)$ values for the designed decay factors with constant background effects as I_i. Through these graphs, it can be clearly seen that the background effects have a successful impact on the cells' enhancement intensity. It can also be

observed that though the background effects vary the intensity of enhancement for every gray level in the image, the final enhancement rate outputted is bound by the global impact pattern defined by the decay factor. Thus, this design of the weights can be concluded as near optimal for our application.

5.3.4 Iteration Number

The final parameter is the number of iterations. Even though the weights and decay factors help the SICNN to automatically orient itself to the varying intensities of an image, the overall desired enhancement would vary from radiologist to radiologist. Every user has his or her own perceptive levels. For this reason, the number of iterations required for the enhancement process is dependent on the user.

The number of iterations defines the amount of enhancement. The number of iterations defines the number of times the network process is to be repeated. This is not the same as scaling or intensity windowing. In scaling, the realized output pattern is only multiplied by a given number to obtain higher clarity.

On the other hand, increasing the number of iterations generates more accurate results, as each iteration repeats the whole network process to enhance more gray levels and generate a completely new output pattern. The system can be seen as a two stage network, where the first stage outputs are used as inputs to calculate the next stage's output. In this process, the original input pattern is used as the initial conditions to generate the first iteration outputs.

5.4 Conclusion

In this chapter, we began by reviewing the working principle of ultrasound technology and its applications in medical therapy and diagnoses. Here, we saw that the speckle patterns produced by the use of coherent transducers in forming the ultrasound images were also a source of contrast resolution degradation.

Speckle patterns that simulate the structure of tissue originated from the constructive interference. Conversely, the destructive ones that bear little resemblance of the actual acoustical tissue microstructure - known as the speckle noise, caused the degradation on the contrast resolution.

The digital SICNN for enhancing the UI images was then explained and its parameters were summarized. Targeting the speckle noise as the core problem in ultrasound imaging, these parameters were designed to solve the ambiguities in the speckle patterns. The decay factor was the main element which controlled the point of maximum enhancement.

An automatically varying weight system, based on the neighbouring effects, was implemented to determine the direction of enhancement and the intensity of the background effects. Though the weights controlled the rate and direction of enhancement, the decay factor had a dominating effect on the global enhancement pattern. Finally, the number of iterations was varied to suit the user's enhancement needs.

Chapter 6

Ultrasound Image Enhancement: Implementation and Analysis

6.1 Introduction

In the earlier chapters, we studied the various image enhancement techniques and the different measures used to quantify the enhancement. The SICNN was studied in greater detail as it is the main system used for our application. The applications of ultrasound imaging and its operating principles were explained. Based on this information, the SICNN system was designed to reduce the ambiguities in the ultrasound speckle patterns. However, the performance of the designed system to highlight the variations in the speckle patterns can only be obtained by testing it on clinical UI images.

This chapter presents the enhancement results of the different techniques on clinical UI images. Section 6.2 gives an overview of the UI databases collected from various hospitals and organisations. This is followed by Section 6.3, where the measures used to quantify the enhancement are explained. Section 6.4 presents the implementation results of some traditional enhancement systems and the SICNN system. Both high and low quality UI images acquired from different equipments are used to examine the networks. These results are comparatively analysed and a combination of systems is suggested to achieve the desired enhancement in both the high and low quality UI cases.

6.2 Data Preparation

All the data used in this study are taken from clinical ultrasound scanners of various hospitals and radiological centres. The outputs of the UI scanners are possibly recorded in three ways: (i) they are burnt onto monochrome positive films (e.g., X-ray films), (ii) they are printed on photographic sheets, (iii) they are shared on digital media, like CD's and local networks. Digital UI patterns are stored in a DICOM format. DICOM formats are used for digital communication in medicine and are a standard regulation set by the American College of Radiologists.

The UI database used in this thesis consists of 3 sets of images with a total of 275 images collected from over 12 hospitals and radiological centres. Though all the institution names cannot be listed here, the main contributions were made by the *Royal Perth Hospital, WA* and *MA Scan and Research Centre, India.*

- *Set* 1 consists of 228 images of DICOM3 format extracted from GE's advanced models; the GE 700 series machines. The size of each image in this set is 630 x 461. The SICNN performance will be mainly accounted for the images in this set.

- *Set* 2 consists of 7 images of size 378 x 283 each, extracted from GE 2244 series ultrasound scanners. The images in this set are selectively picked to demonstrate their lesser quality in comparison to those from more advanced UI scanners.

- *Set* 3 consists of 22 Doppler UI images from the GE 700 series scanners and 18 UI images re-scanned from monochrome films using HP1200dpi scanners. Because of the loss of information in the re-scanned images and the use of colour patterns in Doppler images, we do not emphasise their enhancement results in later sections.

Since DICOM format images are not commonly supported by all softwares, they have been converted to JPEG format. As it is a privacy regulation of most institutions, the patient credentials from the acquired images have been manually edited and deleted using Microsoft Paint software. However, this does not affect the size or even the actual

ultrasound patterns of the image.

The final part of the data preparation is to create a database for the Region of Interest (ROI) of the images. Every UI image contains the actual ultrasound patterns in the centre with the image details to its boundaries. The actual ultrasound patterns are regarded as the ROI. Though the total size of most of the images could be the same, the ROI of every image is generally different. Using MATLAB 5.1 we created the ROI database for all 275 images by manually selecting the ROI of each image.

6.3 Performance Measures

During implementation, the whole image is used as the input to the network. However, the performance measures are applied only to the ROI. The quantifiable performance of the network is based on the contrast variations and the dynamic range measures.

The dynamic range is given by the difference between the maximum and minimum intensities of the ROI.

$$DR = I_{\max} - I_{\min} \tag{6.1}$$

where DR is the dynamic range, I_{max} is the maximum intensity in the ROI and I_{min} is the minimum intensity in the ROI of a given image.

We have seen some of the common contrast measures in Chapter 2. The contrast improvement index, CII, only measures the enhancement of the background intensities in a small window centred on the pixel – it does not take into account the enhancement of the edge pixels relative to the background. Conversely, the gradient enhancement measure, GEM, only measures the enhancement of the edge pixel's gradient, with no account of the change in the background intensity. The relative edge enhancement, REE, measure takes into account both the edge response and the background intensities; however, it can increase without bounds.

Because of the divergences in the pixel enhancement and its background, a measure that is more appropriate is desired. This can be achieved by modifying the Michelson contrast

formula, to calculate the contrast between the enhanced pixels and their corresponding backgrounds.

For a given neighbourhood, let I_{ik} be the intensity of the pixel under consideration and b_{ik} be the average background pixel intensity computed over the neighbourhood. The contrast score c_{ik} for pixel (i, k) is defined as:

$$c_{ik} = \frac{|I_{ik} - b_{ik}|}{|I_{ik} + b_{ik}|} \tag{6.2}$$

When comparing the contrasts of the enhanced and the original images, an extension to the above formula can be used:

$$RCI = \frac{c_{ik}(y)}{c_{ik}(x)} \tag{6.3}$$

where RCI is the relative contrast index, $c_{ik}(y)$ is the contrast of the output pixel and $c_{ik}(x)$ is the contrast of the corresponding input pixel.

Through this method, the contrast increment or decrement can be determined. If the $RCI > 1$ then the contrast has increased and if the $RCI < 1$ then the contrast has decreased.

6.4 Performance Analysis

The performance analysis is the final stage of the thesis. Here the performances of various enhancement schemes are tested and analysed. We analyse the results based on both their quantifiable statistics and their visual suitability. Visual inspection is a very subjective procedure and could vary from person to person. However, we only discuss the highly evident characteristics of the outputs, like the visual clarity of pattern discontinuities in UI images.

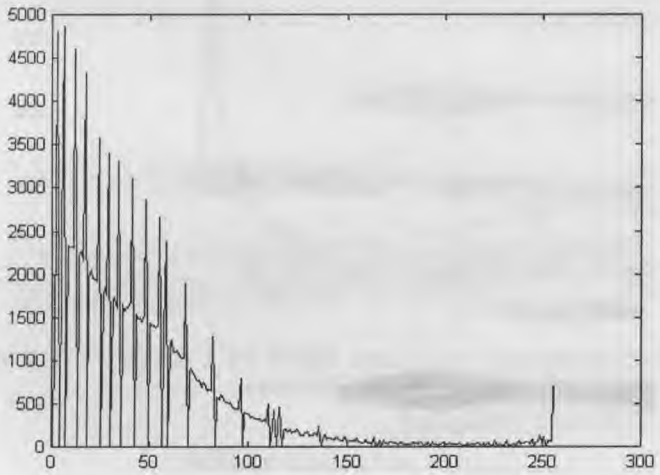
We begin the analysis by testing performance of some commonly used techniques like Histogram Equalisation, Logarithmic Transformation, and Linear Stretching. This is given in Sections 6.4.1 to 6.4.3, respectively. Section 6.4.4 deals entirely with the performance of the designed SICNN enhancer. Based on the analysis of these techniques,

Section 6.4.5 suggests a combination of systems that generates consistent outputs for all the test images.

All the systems are tested on both high and low quality images. Set 1 (database 1) contains images with higher image statistics and Set2 (database 2) contains selectively picked images demonstrating the possible quality degradations. *Figure* 6.1 and *Figure* 6.2 show samples of images picked from Set 1 and Set 2 respectively. These images would be consistently used for all future analysis.

(*a*)

(*b*)



Dynamic Range : 255

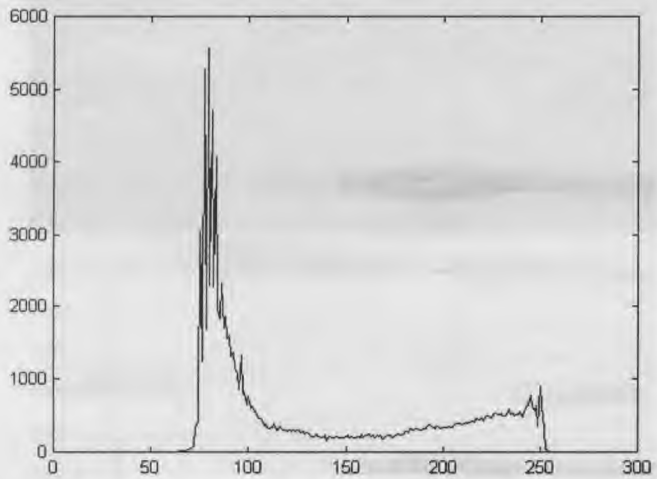
Contrast Score : 28.1066

Figure 6.1 *(a) Set 1 UI image and (b) its histogram plot.*

By quantifying the image statistics, the Set 1 image shows a high DR of 255 and a contrast score of 28.1066. On the other hand, the Set 2 image has lower statistics with a DR of 202 and a contrast score of 13.3485. Note that, the contrast score is an average of the individual pixel contrasts computed over the ROI.

(*a*)

(*b*)



Dynamic Range : 202

Contrast Score : 13.3485

Figure 6.2 *(a) Set 2 UI image and (b) its histogram plot.*

6.4.1 Histogram Equalisation

Histogram Equalisation operates by plotting the output histogram to be uniform over a desired intensity range (I'_{min}, I'_{max}). Assuming I is the intensity of the pixel under

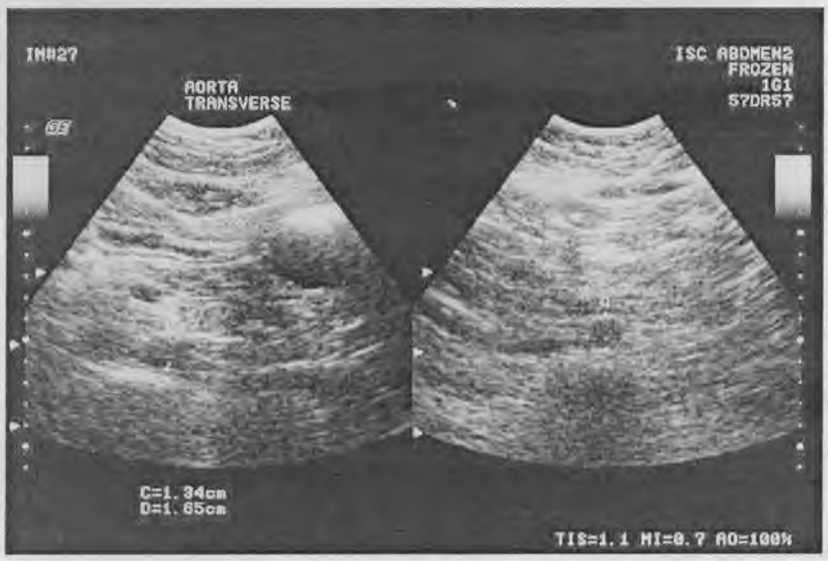
consideration, I_{min} is the minimum input intensity, N is the total number of pixels and n_i is the number of pixels in the image with gray level c, the new pixel intensity I' is given by:

$$I' = \frac{I'_{\max} - I'_{\min}}{N} \sum_{c=I_{\min}}^{I} \frac{n_i}{N} + I_{\min} \tag{6.4}$$

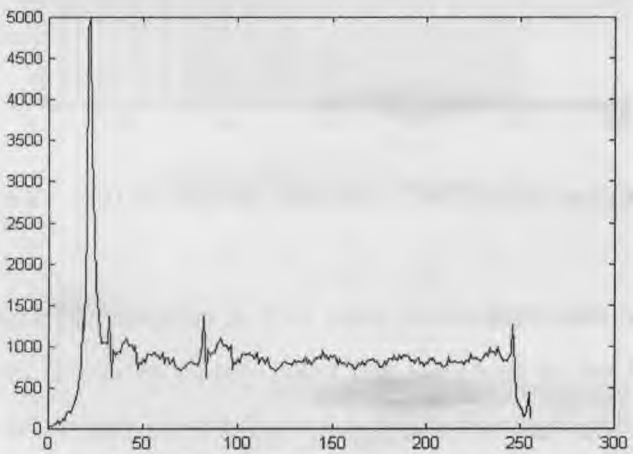
Set 1 *Images*:

Figure 6.3 shows a histogram equalized Set 1 image. As the (I'_{min}, I'_{max}) values for the experiment were given as (0, 255), the resulting DR is 255. The contrast score has increased to 30.8984, showing an RCI variation of 19.98% against the input.

(*a*)



(*b*)



Dynamic Range : 255

Contrast Score : 30.8984

RCI (*input-output*): 1.1998

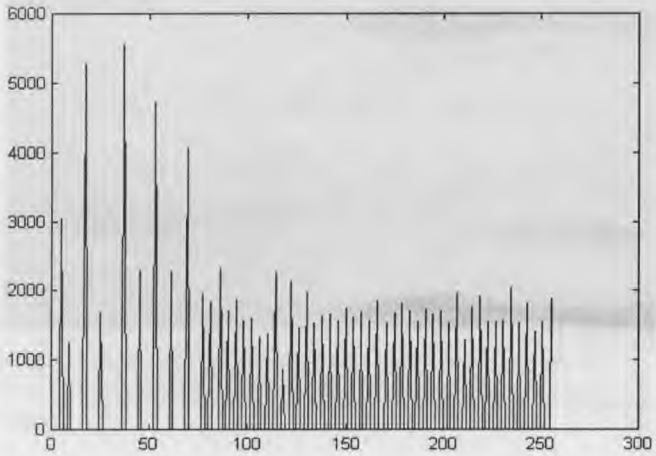
Figure 6.3 *(a) Histogram Equalised Set* 1 *image and (b) its histogram plot.*

***Set 2 Images*:**

The network performance for this set of images has shown reasonable results with a DR of 255 and a contrast score of 16.6844. The RCI shows a high enhancement of 71.9%.

(*a*)

(*b*)



Dynamic Range : 255

Contrast Score : 16.6844

RCI (*input-output*): 1.7190

Figure 6.4 *(a) Histogram Equalised Set 2 image and (b) its histogram plot.*

The network performance in both cases shows high rates of enhancement. But on visual observation, it can be clearly noted that variations in the UI patterns have decreased in both sets of images. As we desire a network that highlights the speckle variations, it can be concluded that this network is not useful to lower the ambiguities in the ultrasound imaging.

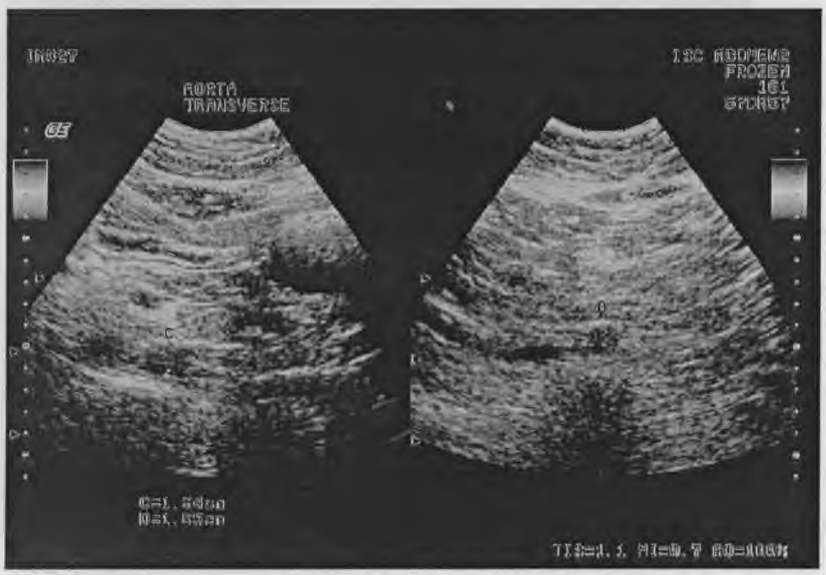
6.4.2 Logarithmic Transformation System

The Log Transformation system operates by calculating the logarithm of the pixel intensities. Using a given threshold t, the log values are accordingly added or subtracted to the input intensity of the pixel.

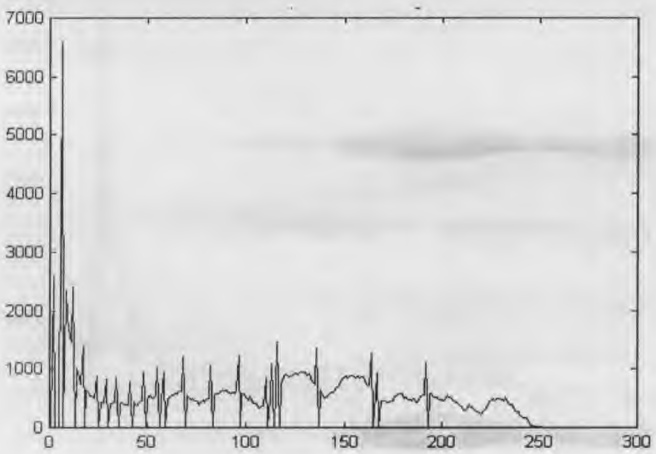
***Set* 1 *Images*:**

Using $t = 128$, the logarithm system is first experimented on the Set 1 image. *Figure* 6.5 shows the system output and its histogram.

(*a*)



(*b*)



Dynamic Range : 255

Contrast Score : 28.9021

RCI (*input-output*): 1.0639

Figure 6.5 *(a) Log transformed Set* 1 *image and (b) its histogram plot.*

The DR of the output is 255. The contrast has increased to 28.9021, showing an enhancement of 6.39% over the input.

***Set 2 Images*:**

Using the same threshold, the system is executed on a Set 2 image. As this image is of a lower quality, the changes in the image quality are more evident here. *Figure* 6.6 shows the output of the log system. The DR of this image has increased to 238. The contrast has also increased to 14.0900, showing an average enhancement of 10.79% over the input.

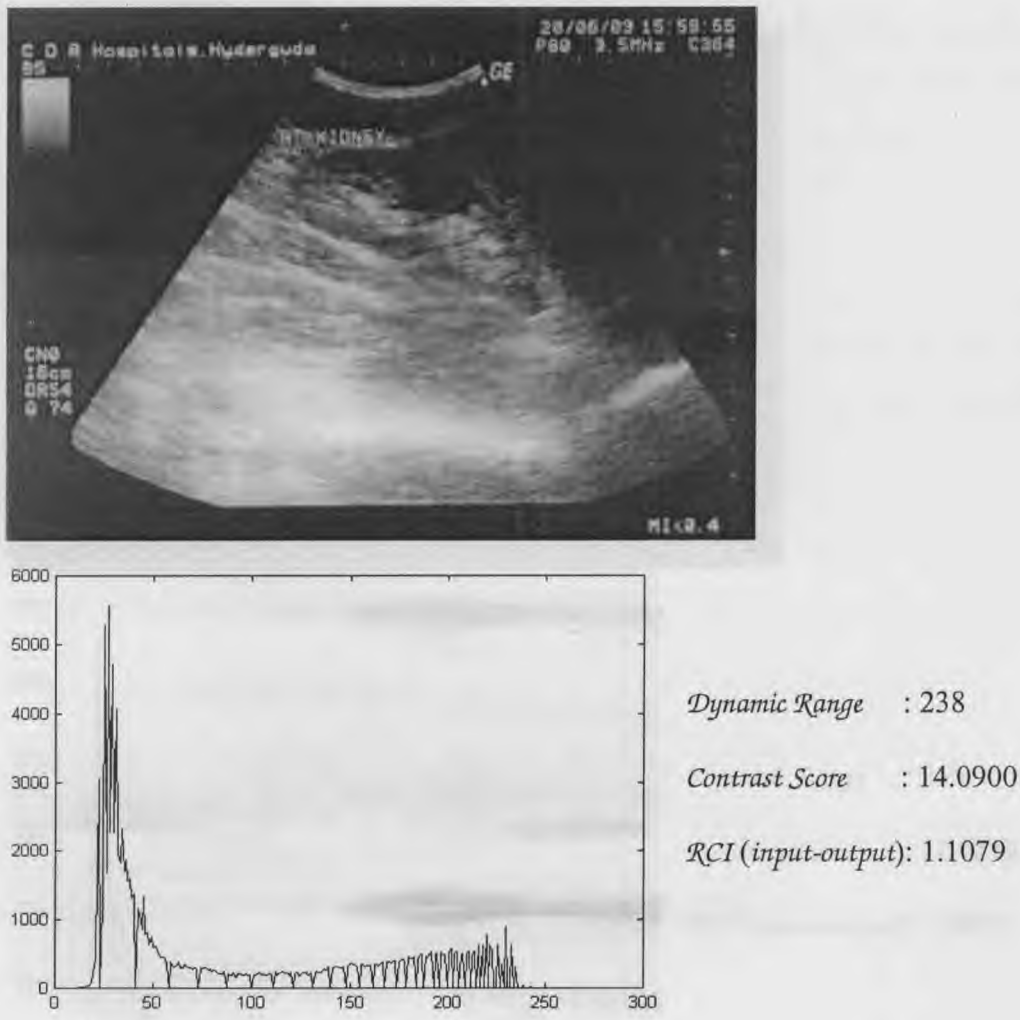


Figure 6.6 *(a) Log transformed Set 2 image and (b) its histogram plot.*

6.4.3 Linear Stretching

In Linear Stretching the transformed pixel intensity I' is obtained by varying the dynamic range of the image. If I_{min}, I_{max} are the minimum and maximum intensities of the image, I is the intensity of the pixel under consideration and $(I'_{min}, I'_{max}) = (0, 255)$:

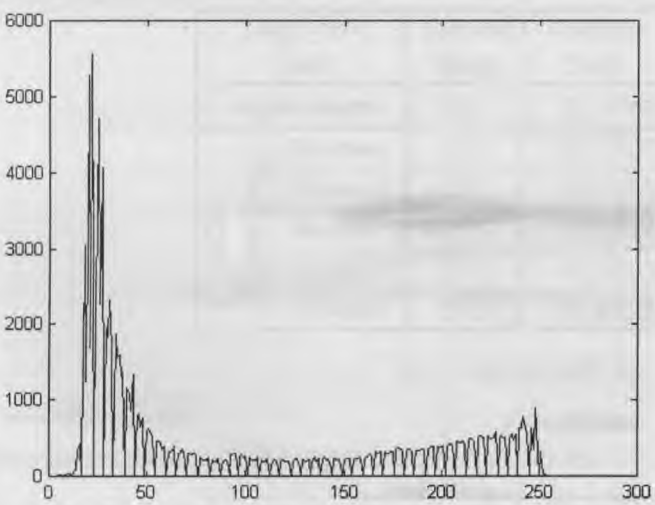
$$I' = I'_{\min} + \frac{(I - I_{\min})}{(I_{\max} - I_{\min})}(I'_{\max} - I'_{\min}) \tag{6.5}$$

***Set 2 Images*:**

Figure 6.7 shows the result of linear stretching a Set 2 image.

(*a*)

(*b*)



Dynamic Range : 255

Contrast Score : 13.9228

RCI (*input-output*): 1.0818

Figure 6.7 *(a) Linearly stretched Set 2 image and (b) its histogram plot.*

The DR of the stretched image is 255. By expanding the DR, the contrast of the image also increases. The contrast score for this stretched image is 13.9228. This is approximately 8% increment in the contrast of the image due to Linear Stretching.

Note that the contrast enhancement using this technique is achieved only by increasing the DR of the image. But since the DR of Set 1 images are already 255, there would be no change in the output statistics of the images.

6.4.4 SICNN System

In this section, we implement the designed SICNN system on Set 1 and Set 2 ultrasound images. The inputs to the system are the original images. In all cases of experimentation, the results of the first five iterations are used to examine the network. However, only the third iteration output images are displayed for comparison.

***Set* 1 *Images*:**

Here, we use *Figure* 6.1 (*a*) as the input to the network. Observing the output statistics in *Table* 6.1 the contrast score of the image has gradually increased with the increase in the iterations, showing an improvement in the contrast of the image.

Table 6.1 *Output statistics of the SICNN system with an input image from Set* 1.

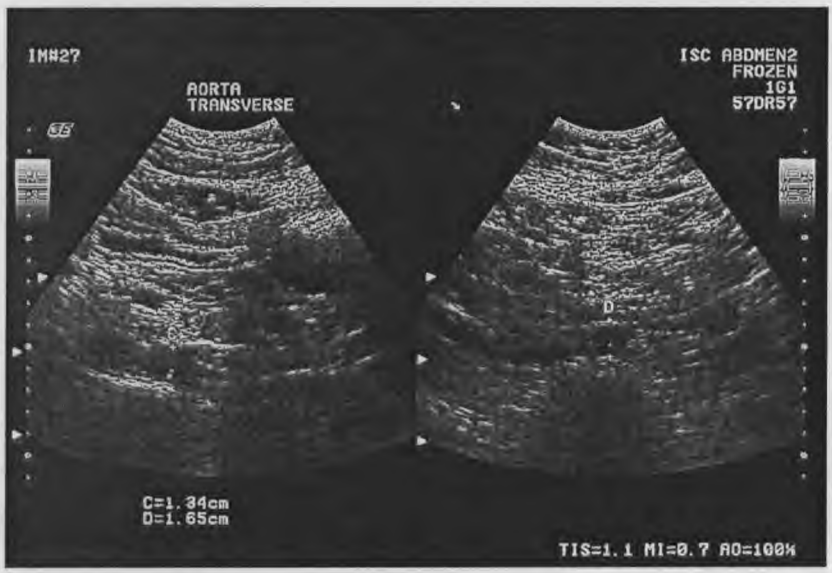
Image from Set1		*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image		255	28.1066	-
SICNN Output Image	*Iteration 1*	255	28.7905	1.0567
	Iteration 2	255	29.6508	1.1565
	Iteration 3	255	30.5634	1.2650
	Iteration 4	255	31.5043	1.3736
	Iteration 5	255	32.4393	1.4816

(*ip* → *input*; and *op* → *output*)

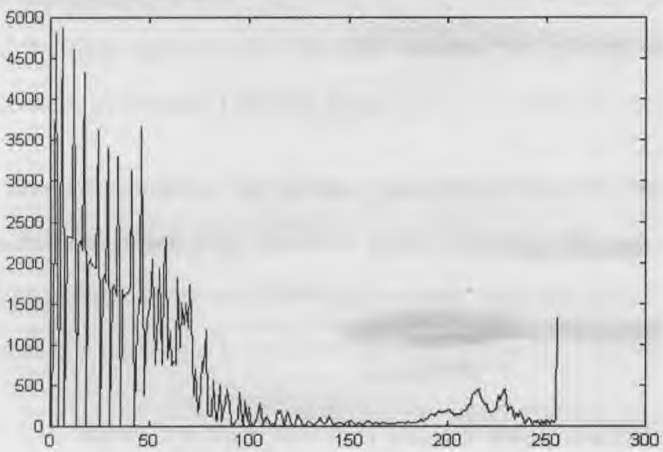
From the characteristics of the input image, it is evident that the dynamic range is already at the highest value and hence cannot increase further. The average RCI computed over the region of interest, between the input and each iteration output, show that an

enhancement of about 5% to 48% is achieved. It should be noted that though the contrast increases with the increase in the iteration number, the required enhancement is subjective to the user's needs and is usually about two to three iterations for Set 1 images. The third iteration output of the SICNN network is given in *Figure* 6.8. From this histogram, it is clear that the gray levels around 130 are enhanced the most. From visual inspection also, it can be seen that the variations in the ultrasound patterns are clearly highlighted.

(*a*)



(*b*)



Dynamic Range : 255

Contrast Score : 30.5634

RCI (input-output): 1.2650

Figure 6.8 *(a) Third iteration SICNN output for the Set* 1 *image, and (b) its histogram.*

Set 2 Images:

From *Table* 6.2, it is clear that the contrast scores of the SICNN output after every iteration have increased considerably. The RCI results of Set 2 images vary from 5% for the first iteration to 70% for the fifth iteration. Upon comparison the rate of enhancement for Set 2 images is higher than that of Set 1 images. However, the increment in the dynamic range of the ROI is only marginal and not as high as Set1 images.

Table 6.2 *Output statistics of the SICNN system with an input image from Set* 2.

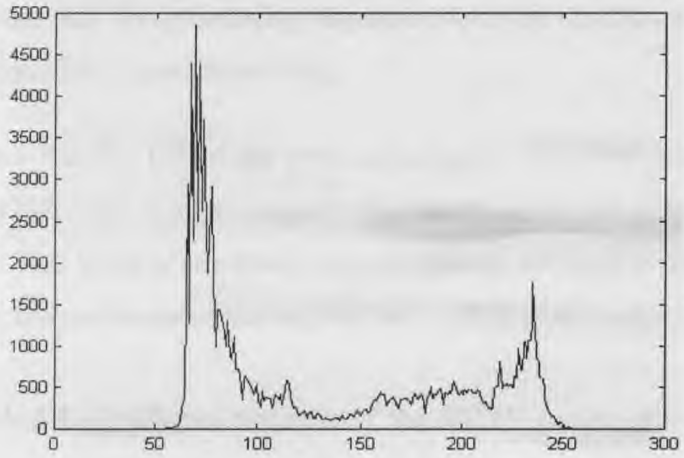
Image from Set2		*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image		202	13.3485	-
SICNN Output Image	*Iteration 1*	202	13.9055	1.0549
	Iteration 2	203	15.0705	1.1970
	Iteration 3	204	16.3959	1.3556
	Iteration 4	205	17.8786	1.5221
	Iteration 5	206	19.4494	1.6978

(*ip* → *input*; and *op* → *output*)

From the histogram of the input image, it is evident that the image is overly bright and has a DR of only 202. But the decay factor of the SICNN was designed for a full DR of 255. It was assumed that the partially detected ultrasound reflections were displayed as the middle order gray levels. Because of this, the system was designed to have maximum enhancement at around 128 and least at the two ends (0 and 255).

In the present situation, the darkest patterns of the UI image are positioned around 100 and the middle order gray levels at approximately 190. As a result, the darker UI patterns are enhanced the most and the middle order gray levels of the image are only marginally enhanced. By logic this is undesirable.

Figure 5.11 shows the third iteration SICNN output whose contrast score has increased to 16.3959, displaying an average RCI of 1.3556. The DR has increased by only two shades to 204.



(*b*)

Dynamic Range : 204

Contrast Score : 16.3959

RCI (input-output): 1.3556

Figure 6.9 *(a) Third iteration SICNN output for the Set 2 image and (b) its histogram.*

6.4.5 SICNN System with Dynamic Range Expansion

In the previous section, we saw the effects of applying the designed SICNN system directly to the UI images. Direct application of the SICNN on Set 1 images did not have any negative effects because all images in this set have a DR of 255 and satisfy the design conditions of the SICNN.

The dynamic ranges of images in Set 2 are all less than 255. Some of the images in this

database were also manually brightened or darkened. Because of this, the decay pattern does not have the same impact as it was designed for. However, this problem can be overcome by simply expanding the dynamic ranges of all the images to 255 before processing the image through the SICNN system.

In this section, we pre-process the input image to expand the dynamic range of any given image. The resulting image is then used as an input to the SICNN system and the new performance scores are analysed.

As seen earlier, all the Set 1 images satisfy the SICNN design conditions. However, the Set 2 images do not satisfy the conditions as they have a low dynamic range. Hence, we choose a system that ensures maximum dynamic range. From analysis of the previous sections, this pre-processing requirement of the SICNN can be satisfied by a simple technique like Linear Stretching.

We saw that the DR of the stretched image is 255. The contrast of this image increased to13.9228. This is approximately 8% increment in the contrast due to Linear Stretching alone. This linearly stretched image is used as an input to the SICNN network. *Table* 6.3 shows the performance statistics of the SICNN system after linear stretching.

Table 6.3 *Output statistics of the SICNN system after linear stretching. The input image is from Set* 2.

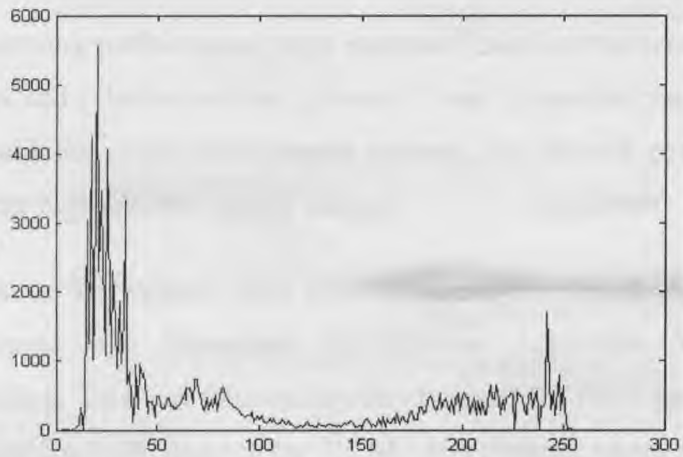
Image from Set2		*Dynamic Range*	*Contrast Score*	*RCI*		
				ip-sp	*sp-op*	*ip-op*
Input Image		202	13.3485	-	-	-
Linearly Stretched Image		255	13.9228	1.0818	-	-
SICNN Output Image	*Iteration 1*	255	14.4851	-	1.0393	1.1353
	Iteration 2	255	15.4907	-	1.1256	1.2256
	Iteration 3	255	16.6683	-	1.2297	1.3344
	Iteration 4	255	17.8805	-	1.3348	1.4436
	Iteration 5	255	19.0889	-	1.4389	1.5510

(*ip→ input image*; *sp→ linearly stretched image*; *op→ output image*)

The contrast scores of this system (*Table* 6.3) are similar to that of the SICNN without stretching (*Table* 6.2). From this, it is clear that by using the pre-processing technique, the designed enhancement pattern is achieved with a higher dynamic range and only a marginal variation in the contrast scores. *Figure* 6.10 shows the third iteration SICNN output after linear stretching.

(a)

(b)



Dynamic Range : 255

Contrast Score : 16.6683

RCI (input-output): 1.3344

Figure 6.10 *(a) Third iteration output of SICNN for a Linearly Stretched Set* 2 *image and (b) its histogram.*

The RCI (ip-op) gives the total enhancement of the network with respect to the input. However, the actual performance of the SICNN alone is given by comparing the network

output to the linearly stretched image. This is given by the RCI (sp-op). By comparing the SICNN performances of *Table* 6.1 and *Table* 6.3, it can be concluded that for both Set 1 and Set 2 images, we achieve approximately 25% variation in the contrast pattern for the third iteration outputs.

It should be noted that all images in Set 1 have a dynamic range of 255. Hence, processing these images through the SICNN system with Linear Stretching will have the same output as the SICNN system without Linear Stretching.

6.5 Conclusion

This chapter dealt with the performance analysis of various enhancement techniques implemented on two dimensional clinical UI images. The diagnostic images collected from various hospitals were categorized into different sets. Set 1 contained all the high quality images and Set 2 contained the lower quality images. The network performances were tested on both sets of images.

The network performances were examined based on the average contrast scores, dynamic ranges and relative contrast indexes. Visual inspection was also used to determine the compatibility of the enhancement systems. All network performances were analysed on both the high and low quality images.

A total of five systems were implemented and analysed. We started with the traditional techniques like Histogram Equalization, Logarithm Transformation and Linear Stretching. This was followed by the designed SICNN's performance on UI images and the combined efficiency of the SICNN with dynamic range expansion.

Though the quantifiable results of the histogram equalization were high, the technique did not show usefulness in highlighting the speckle variations of the UI images. The dynamic range expansion of linear stretching showed about 8% contrast enhancement for Set 2 images. However, the technique was not useful for Set 1 images as all Set 1 images already had a dynamic range of 255. The log transformation showed about 6%

enhancement for Set 1 images and 10% enhancement for Set 2 images. But the dynamic ranges of the log transformed outputs were not as consistent as that of the linearly stretched outputs.

The SICNN design showed promising results for Set 1 images, but had problems with Set2 images. The drawbacks of the designed SICNN in enhancing low quality images were highlighted. It was observed that these drawbacks could be overcome by maximising the dynamic range of the image before processing it through the SICNN. Linear stretching technique was used to achieve this. This new network demonstrated reasonable enhancement for both Set 1 and Set 2 images. An average SICNN enhancement of about 25% in the third iteration was achieved, for any given image.

References

Abdou, I.E., and Pratt, W.K. (1979). "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors." *Proceedings of the IEEE*, vol.67, no. 5, pp. 753-763, May.

Arik, S. and Tavsanoglu, V. (1996) "Equilibrium Analysis of non-symmetric CNN's" *International Journal on Circuit Theory and Applications: Special issue on Cellular Neural Networks,* vol.24, pp. 269-674.

Barlow, H.B. (1981). "The Ferrier lecture. Critical Limiting Factors in the Design of the Eye and the Visual Cortex. " *Proc. R. Soc. Lond.,* vol. B212, pp. 1-34.

Barone, A., Balsi., M. and Cimgalli, V. (1993). "Polynomial Cellular Neural Networks: A New Dynamical Circuit For Pattern Recognition." *Proceedings of International Specialist Workshop on Nonlinear Dynamics of Electronic Systems*, July. Dresden, Germany.

Beghdadi, A. and Le Negrate, A. (1989) "Contrasts Enhancement Technique based on Local Detection of Edges" *Computer, Vision, Graphics and Image Processing*, vol. 46, pp.162-174.

Bouzerdoum, A. (1991) "Nonlinear Lateral Inhibitory Neural Networks: Analysis and Application to Motion Detection" *PhD thesis*, University of Washington, July.

Bouzerdoum, A. (1994) "A Hierarchical Model for Early Visual Processing" *Proceedings of SPIE on Human Vision, Visual Processing, and Digital Display V,* vol. 2179,pp. 10-17, 8-10 February, San Jose, California, USA.

Bouzerdoum, A., and Pinter, R.B. (1993) "Shunting Inhibitory Neural Networks: Derivation and Stability Analysis" *IEEE Transactions on Circuits and Systems-I,* vol.40, no.3, pp. 215-221, March.

Brodie, S.E., Knight, B.W. and Ratliff, F. (1978). "The Spatiotemporal Function of the Limulus lateral Eye." *J. General Physiology,* vol72, pp. 167-202.

Cheng C-H. and Wu, C-Y. (2002) "The Design of Ratio Memory Cellular Neural Network (RMCNN) with Self-Feedback Template Weight for Pattern Learning and Recognition" *Cellular Neural Networks and their Applications (CNNA 2002): Proceedings of the 2002 7th IEEE International Workshop.* pp. 609-615, 22-24 July.

Cheung, H. N., Bouzerdoum, A., and Newland W. (1999) "Properties of Shunting Inhibitory Cellular Neural Networks for Colour Image Enhancement" *Neural Information Processing, 1999. Proceedings of the 6th ICONP International Conference,* vol. 3, pp. 1219-1223, 16-20 November. Edith Cowan University, Joondalup, Australia.

Chua, L.O., and Roska T., (1993) "The CNN Paradigm" *IEEE Transactions on Circuits and Systems-I,* vol. 40, no.3, pp. 147- 156, March.

Chua, L.O., and Roska T., (2002) "*Cellular Neural Networks and Visual Computing: Foundations and Applications*" pp 114. University Press, Cambridge, U.K.

Chua, L. O. and Wu, C.W. (1992) "On the Universe of Stable Cellular Neural Networks" *International Journal on Circuit Theory and Applications: Special issue on Cellular Neural Networks,* vol.20, pp. 497-517, July-August.

Chua, L.O. and Wu, C.W. (1997) "A More Rigorous Proof of Complete Stability of Cellular Neural Networks" *IEEE Transactions on Circuits and Systems,* vol. 44, no. 4, pp. 370-371, April.

Chua, L.O., and Yang L., (1988) "Cellular Neural Networks: Theory". *IEEE Transactions on Circuits and Systems*, vol. 35, no.10, pp. 1257- 1272, October.

Chua, L.O., and Yang L., (1988) "Cellular Neural Networks: Applications". *IEEE Transactions on Circuits and Systems*, vol. 35, no.10, pp.1273- 1290, October.

Cimagalli, V., Bobbi, M., and Balsi, M. (1993) "MODA: Moving Object Detecting Architecture." *IEEE Transactions on Circuits and Systems-II,* vol. 40, no. 3, pp. 174-183, March.

Civalleri, P.P. and Gilli, M. (1999) "On Stability of Cellular Neural Networks", *Journal for VLSI Signal Processing*, Kluwer Academic Publisher, vol.23, no.2/3, pp. 429-435.

Corinto, F., Gilli, M., and Civalleri, P.P. (2002) "On Stability of Full Range and Polynomial Type CNNs" *Cellular Neural Networks and their Applications: Proceedings of the 2002 7th IEEE International Workshop,* pp. 33-40, 22-24 July.

Crounse, K.R., Roska, T., and Chua, L.O. (1993) "Image Half-toning with Cellular Neural Networks" *IEEE Transactions on Circuits and Systems-II,* vol. 40, no. 4, pp. 267-283, April.

Crounse, K.R. and Chua, L.O. (1995) "Methods for Image Processing and Pattern Formation in Cellular Neural Networks, A Tutorial" *IEEE Transactions on Circuits and Systems-I,* Special Issue on Nonlinear Waves, Patterns and Spatio-Temporal Chaos in Dynamic Arrays, vol. 42, no. 10, pp. 583-601, October.

Cruz, J.M. and Chua, L.O. (1991) "A CNN Chip for Connected Component Detection" *IEEE Transactions on Circuits and Systems,* vol. 38, no. 7, pp. 812-817, July.

Dash, L. and Chatterji, B.N, (1991). "Adaptive Contrast enhancement and De-enhancement." *Pattern Recognition,* vol. 24, no. 4, pp. 289-302.

Dellart, F., and Vandevalle, J. (1994). "Automatic design of Cellular Neural Networks by means of Genetic Algorithms- Finding a Feature Detector." *Cellular Neural*

Networks and their Applications CNNA-94., Proceedings of the 3rd IEEE International Workshop. pp. 189-194, 18-21 December. Rome, Italy.

Deutsch, S. and Deutsch, A. (1993). *Understanding the Nervous System. An Engineering Perspective.* IEEE Press.

Dhawan, A.P., Beulloni, G., and Gordon, R. (1986) "Enhancement of Mammographic Features by Optimum Adaptive Neighbourhood Image Processing." *IEEE Transactions On Medical Imaging,* vol. 5, no. 1, pp. 8-15, March.

Ecimovic, P. and Wu, J. (2002) "Delay-Driven Contrast Enhancement Using Cellular Neural Network with State-Dependent Delay" *Cellular Neural Networks and their Applications: Proceedings of the 2002 7th IEEE International Workshop.* pp. 202-208, 22-24 July.

Galias, Z. (1993). "Designing Cellular Neural Networks for the Evaluation of Local Boolean Functions." *IEEE Transactions on Circuits and Systems-II,* vol.40, no. 3, pp. 219-222, March.

Gilli, M. (1994) "Stability of Cellular Neural Networks and Delayed Cellular Neural Networks with Non-positive Templates and Non-Monotonic Output Functions" *IEEE Transactions on Circuits and Systems-I,* vol. 41, no. 8, pp. 518-528, August.

Gonzalez, R.C., and Woods, R. E., (2001) "*Digital Image Processing*", pp. 162-218, Sixth Indian Reprint, Addison- Wesley Longman, Delhi, India

Goras, L., Chua, L.O. and Leenaerts, D.M.W. (1995) "Turing Patterns in CNN's- Part I: Once Over Lightly" *IEEE Transactions on Circuits and Systems-I,* vol. 42, pg 602-611, October 1995.

Goras, L. and Chua, L.O. (1995) "Turing Patterns in CNN's- Part II: Equations and Behaviours" *IEEE Transactions on Circuits and Systems-I,* vol. 42, pp. 612- 626, October.

Goras, L., Chua, L.O. and Pivka, L. (1995) "Turing Patterns in CNN's- Part III: Computer Simulation Results" *IEEE Transactions on Circuits and Systems-I,* vol. 42, pp. 627- 636, October.

Goras, L., Ghinea, R., Teodorescu, T.D. and David, E. (2002) "On the Dynamics of a Class of Cellular Neural Networks" *Cellular Neural Networks and their Applications: Proceedings of the 2002 7th IEEE International Workshop,* pp. 92-97, 22-24 July.

Gordon, R. and Rangayyan, R.M. (1984) "Feature Enhancement of Film Mammograms Using Fixed and Adaptive Neighbourhoods" *Applied Optics,* vol. 23, no. 4, pp. 560-564, February.

Guillon, S., Baylou, P. and Najim, M. (1996) "Robust Nonlinear Contrast Enhancement Filters." *Proceedings of IEEE International Conference on Image Processing,* vol. I, pp. 757-760. Lausanne, Switzerland.

Harrer, H. (1993). "Multiple Layer Discrete-Time Cellular Neural Networks Using Time Variant Templates." *IEEE Trans. On Cicuits and Systems-II,* vol. 40, no.3, pp. 191-199, March.

Harrer, H. and Nossek, J.A. (1992). "Discrete Time Cellular Neural Networks." *Int. J. Circuit Theory and Applications,* vol.16, no.5, pp. 1268-1271, May.

Harris, J.L. (1977). "Constant Variance Enhancement: A Digital Processing Technique." *Applied Optics,* vol. 16, no. 5, pp. 1268-1271, May.

Hartline, H.K. and Ratliff, F. (1957). "Inhibitory Interactions of Receptor Units in the eye of Limulus." *J. General Physiology,* vol,40, pp. 357-376.

Hartline, H.K. and Ratliff, F.(1958). "Spatila Summations of Inhibitory Influences in the Eye of Limulus and the Manual Interactions of Receptor Units." *J. Genarl physiology,* vol. 41, pp. 1049-1066.

Iannella, N. and Bouzerdoum, A.(1996) "Time Evolution of Receptive Fields" *Proceedings of the 1996 Australian New Zealand Conference on Intelligent Information Systems,* pp.105-108, 18-20 November. Adelaide, South Australia, Australia.

Jernigan, M.E., and McLean, G.F. (1992). *Lateral Inhibition and Image Processing,* chapter 17, pp. 451-463. CRC Press, Boca Raton, USA.

Kawabata, H., Nanba, M. and Zhang, Z. (1997) "On the associative memories in cellular neural networks" *IEEE International Conference on Systems, Man and Cybernetics*, 1997. *Computational Cybernetics and Simulation,* 1997, vol. 1, pp.929-933.

Laiho, M., Paasio, A., and Halonen K. (2000) "Structure of a CNN cell with Linear and Second Order Polynomial Feedback Terms" *IEEE Workshop on Cellular Neural Network and their Applications,* pp.401-405, May. Catania, Italy.

Laine, A.F., Schuler, S., Fan, J., and Huda, W. (1994) "Mammographic Feature Enhancement by Multi-scale Analysis." *IEEE Transactions on Medical Imaging,* vol. 13, no. 4, pp. 725-740, December.

Lan, J-F. and Wu C-H. (1995) "Analog CMOS current-mode implementation of the feedforward neural network with the on-chip learning and storage" *International Symposium on Circuits and Systems*, vol. III, pp. 1676-1679. Seattle, USA.

Lan, J-F. and Wu C-H. (1995) "CMOS current-mode outstar neural networks with long period analog ratio memory" *IEEE International Conference on Neural Networks,* Nov. 27- Dec. 1., Perth, Australia.

Lewis, C.E. (1999) "Ultrasound" [Computer Software]. *Microsoft Encarta'99 Encyclopedia.* Microsoft Corporation.

Linan, G., Espejo, S., Dominguez-Castro, R. and Rodriguez-Vazquez, A.(2000) "The CNNUC3: An Analogy I/O 64x64 CNN Universal Chip Prototype with 7-bit Analog Accuracy " *Proceedings of the Sixth IEEE International Workshop on Cellular Neural Network and Their Applications,* pp. 201-206, May.

Liu, D. and Michel, A.N. (1993) "Cellular Neural Networks for Associative Memories" *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing,* vol. 40, no.2 , pp.119-121, February.

Lukianiuk, A. (1996) "Capacity of Cellular Neural Networks as associative memories" *Fourth International IEEE Workshop on Cellular Neural Network and their Applications,* 1996. CNNA-96. Proceedings, pp.37-40.

Mach, E. (1886a). "Uber die physiologische Wirkung raumlich verteilter Lichtreize III." *Sitzungsberichte der mathematics-naturwissenshaftilichen Classe der kaiserlichen Akademie der Wissenschaften,* vol.54 II, no.134, pp. 131-144.

Mach, E. (1886b). "Uber die physiologische Wirkung raumlich verteilter Lichtreize III." *Sitzungsberichte der mathematics-naturwissenshaftilichen Classe der kaiserlichen Akademie der Wissenschaften,* vol.54 II, no.134, pp. 393-408.

Marr, D. and Hildreth, E.C. (1980). "Theory of Edge Detection." *Proc. R. Soc. Lond. B,* vol.207, pp. 187-217.

Matsumoto, T., Chau, L.O., Suzuki, H. (1990a). "CNN Cloning Template: Hole-Filler." *IEEE Trans, On Circuits and Systems,* vol. 37, no. 5, pp. 635-638, May.

Matsumoto, T., Chau, L.O., and Suzuki, H. (1990b) "Image Thinning with a Cellular Neural Network" *IEEE Trans, On Circuits and Systems,* vol. 37, no. 5, pp. 638-640, May.

Matsumoto, T., Chau, L.O., and Suzuki, H. (1990c) "CNN Cloning Template: Hole-Filler" *IEEE Trans, On Circuits and Systems,* vol. 37, no. 5, pp. 633-635, May.

Matsumoto, T., Chau, L.O., and Suzuki, H. (1990d) "CNN Cloning Template: Hole-Filler" *IEEE Transactions on Circuits and Systems,* vol. 37, no. 8, pp. 1070-1073, August 1990.

Ogata, K., (1987) "*Discrete-time Control Systems*" Prentice- Hall, New Jersey.

Oppenheim, A.V., Schafer, R.W. and Stockham Jr, T.G. (1968). "An Adaptive Motion Decision System for Digital Image Stabalizer based on Edge Pattern Matching." *IEEE Trans. on Consumer Electronics,* vol.38. no.3, pp. 607-615, August.

Pal, S.K. (1982) "A Note on Quantitative Measure of Image Enhancement through Fuzziness" *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 4, no. 2, pp. 204-208, March.

Paradis, M.A.K., and Jernigan, M.E. (1994) "Homomorphic vs. Multiplicative Lateral Inhibition Models for Image Enhancement" *IEEE International Conference on Syst., Man, and Cybernetics*, pp. 286-291.

Paranjape, R.B., Morrow, W.M., and Rangayyan, R.M. (1992) "Adaptive Neighbourhood Histogram Equalisation for Image Enhancement" *CVGIP: Graphical Models and Image Processing,* vol. 5, no. 3, pp. 259-267, May.

Peli, E. (1990). "Contrast in Complex Images" *Journal of the Optical Society of America,* vol.7, no.10, pp.2032-2040, October.

Perez-Munuzuri, V., Perez-Villar, V., and Chua, L.O. (1993) "Auto-waves for Image Processing on a Two-Dimensional CNN array of the Excitable Nonlinear Circuits: Flat and Wrinkled Labyrinth." *IEEE Transactions on Circuits and Systems-I,* vol. 40, no. 3, pp.174-181, March.

Pinter, R.B. (1983a). "Product Term Nonlinear Term Lateral Inhibition and Consequences for Wide Field textured Stimuli." *J. Theoretical Biology,* vol.100, pp. 525-531.

Pinter, R.B. (1983b). "The Electrophysiological Bases for Linear and Nonlinear Product Term Lateral Inhibition and Consequences for Wide Field Textured Stimuli" *J. Theor, Biology*, pp. 233-243.

Pinter, R.B. (1984). "Adaptation of Receptive Field Spatial Organization via Multiplicative Lateral Inhibition" *J. Theoretical Biology,* vol.110, pp. 435-444.

Pinter, R.B. (1985). "Adaptation of Spatial Modulation Transfer Functions via Nonlinear Lateral Inhibition" *Biological Cybertenics,* vol.51, pp. 285-291.

Pizer, S.M., Zimmerman J.B. and Staab, E.V. (1984). "Adaptive grey level assignment in CT scan display" *J. Comput. Assist. Tomogr.,* vol. 8, pp. 300-308.

Pizer, S.M., Austin, J.D., Perry, J.R. and Zimmerman J.B. (1986). "Adaptive Histogram Equalization for automatic contrast enhancement of medical images" *In Proceedings of the 4th International conference on Picture Archiving and Communication System (PACSIV) for Medical Application,* SPIE, 1986, vol. 626.

Pizer, S.M., Amburn, E.P., Austin, J.D. et al. (1987). "Adaptive Histogram Equalization and its Variations" *Computer, Vision, Graphics and Image Processing,* vol. 39, pp. 355-368.

Pontecorvo, C. (1998). "*Edge Detection and Enhancement Using Shunting Inhibitory Cellular Neural Networks*" Unpublished doctoral dissertation, University of Adelaide, Adelaide, South Australia.

Pontecorvo, C. and Bouzerdoum, A. (1995) "Edge Detection in Multiplicative Noise Using the Shunting Inhibitory Cellular Neural Network" *Digital Image Computing: Techniques and Applications,* pp. 637-642, Brisbane, Australia, 6-8 December.

Pontecorvo, C. and Bouzerdoum, A. (1997) "Edge Detection in Multiplicative Noise Using the Shunting Inhibitory Cellular Neural Network" *Proceedings of 1997 International Conference on Engineering applications of Neural Networks,* pp. 281-285, Stockholm, Sweden, 16-18 June.

Proakis, J.G., and Manolakis, D.G. (1992) "*Digital Signal Processing: Principles, Algorithms and Applications*" 2nd edition. Macmillan Publishing Company, New York.

Ratliff, F., Hartline, H.K, and Miller, W.H. (1963). "Spatial and Temporal Aspects of Retinal Inhibitory Interactions." *J. Opt. soc. Am.,* vol. 53, pp. 110-120.

Roksa, T. (1998) "Analogic, CNN Computing: Architectural, Implementation and Algorithmic Advances-A Review" *Proceedings of the Fifth IEEE International Workshop on Cellular Neural Network and Their Applications*, pp. 3-10, 14-17 April. London.

Roksa, T, Boros, T., Radvanyi, A., Thiran, P. and Chua, L.O. (1992a) "Detecting Moving and Standing Objects Using Cellular Neural Networks" *International Journal on Circuit Theory and Applications*, vol. 20, pp. 613-628.

Roksa, T., and Chua, L.O. (1992b) "Cellular Neural Networks with Nonlinear and Delay-Type Elements and Non-Uniform Grids" *International Journal on Circuit Theory and Applications,* vol. 20, pp. 469-481.

Roksa, T., Chua, L.O., Wolf, D., Kozek, R., Tetzlaff, R. and Puffer, F. (1995) "Simulating Nonlinear Waves and Partial Differential Equations via CNN- Part I: Basic Technique" *IEEE Transactions on Circuits and Systems-I,* vol. 42, pp.807-815, October.

Sanderfur, J.T., (1990) "*Discrete Dynamical Systems: Theory and Applications*" Clarendon Press.

Sheu, B.J., Cho K-B., and Young, W.C.(1998) "Integration of Sensor/Processor under Cellular Neural Networks Paradigm for Multimedia Applications" *Proceedings of the Fifth IEEE International Workshop on Cellular Neural Network and their Applications,* pp. 45-49, 14-17 April. London, U.K.

Shi, B.E., Roksa, T., and Chua, L.O. (1993) "Design for Linear Cellular Neural Networks for Motion sequence Filtering" *IEEE Transactions on Circuits and Systems-II,* vol. 40, no.5 , pp.320-321, May.

Slot, K. (1992) "Cellular Neural Networks Design for Solving Specific Image Processing Problems" *International Journal on Circuit Theory and Applications*, vol. 20, pp. 629-637.

Sonka, M., Hlavac, V. and Boyle, R. (1993). *Image Processing, Analysis and Machine Vision.* Chapman and Hall.

Srinivasan, M.V., Laughlin, S.B., and Dubs, A. (1982). "Predictive Coding: a Fresh View of Inhibition in the Retina." *Proc. R. Soc. Lond.,* vol.B216, p.427-459.

Suzuki, H., Matsumoto, T. and Chua, L.O (1992). "A CNN Handwritten Character Recogniser." *Int. J. Circ. Th. Appl.,* vol.20, pp. 601-612.

Sziranya, T. and Csicsvari, J. (1993) "High Speed Character recognition using a Dual Cellular Neural Network Architecture (CNND)" *IEEE Transactions on Circuits and Systems-II,* vol. 40, no. 3, pp. 223-231, March.

Takahashi, N. and Chua, L.O. (1998) "On the Complete Stability of Nonsymmetric Cellular Neural Networks." *IEEE Transactions on Circuits and Systems-I,* vol. 45, no. 7, pp. 754-758, July.

Tetzlaff, R., Kunz, R., Ames, C. and Wolf, D. (1999) "Analysis of Brain Electrical Activity in Epilepsy with Cellular Neural Networks (CNN)" *ECCTD'99- Proceedings of the European Conference on Circuit Theory and Design,* pp. 1007-1010. Stressa, Italy.

Wu, C-Y. and Cheng C-H. (1997) "The Design of CMOS modified Hopfield Neural Network for Pattern Recognition" *International Symposium on Multimedia Information Proceeding,* Session O, pp. 585-590.

Zimmerman, J.B., Pizer, S.M., Staab, E.V., Perry, J.R., Mc Cartney, W., and Brenton, B.C. (1988) "An evaluation of the Effectiveness of Adaptive Histogram Equalisation for Contrast Enhancement" *IEEE Transactions on Medical Imaging*, vol. 7, no. 4, pp. 304-312, December.

Internet References

Fraser, N. (September 21, 1998). *The Biological Neuron.* Retrieved on April 15, 2004, from URL http://vv.carleton.ca/~neil/neural/neuron-a.html

No Author. *Cellular Neural Networks.* Retrieved on April 22, 2004, from URL http://www.ce.unipr.it/pardis/CNN/cnn.html#InterPoint

No Author. *Architecture of Cellular Neural Networks.* Retrieved on May 4, 2004, from URL http://www.isi.ee.ethz.ch/~haenggi/CNN_web/architecture.html

No Author. *Cellular Neural Networks (CNNs): modelling and applications.* Retrieved on May 6, 2004, from URL http://www.scg.dees.unict.it/activities/cnn.html

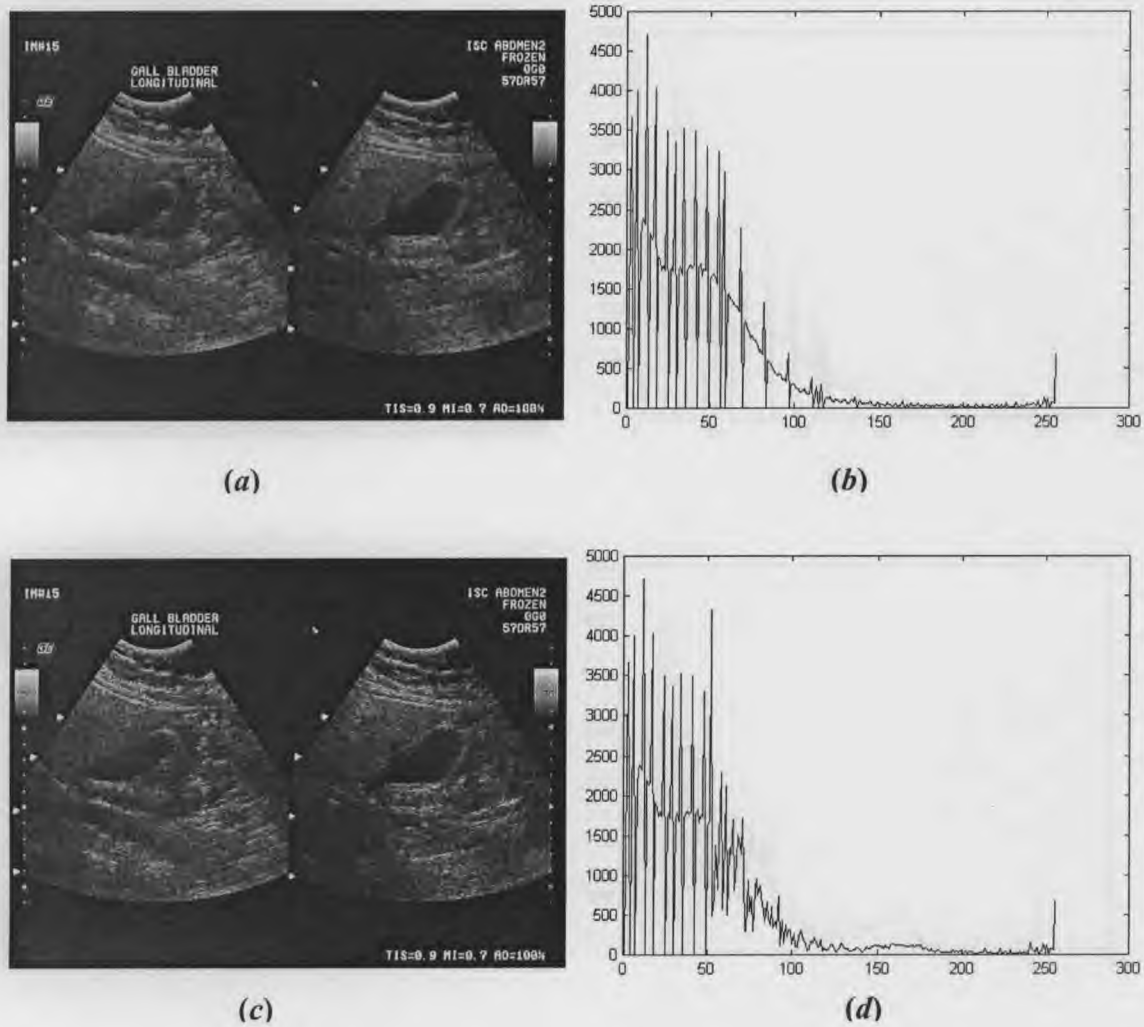
Appendix A

```
%        I N D E X
%ip = matrix containing the input
%(j,1) = size of the neighbourhood
%r = radius of the neighbourhood
%z = neighbour under consideration
%nz = sum of negative neighbours
%pz = sum of positive neighbours
%nzc = negative neighbour count
%pzc = positive neighbour count
%mnz = mean of negative neighbours
%mpz = mean of positive neighbours
%dir = direction of enhancement
%bf = net background effects
%df = net decay factor (a_i)

%        A L G O R I T H M   for Background Effects
i=ip(rows,columns);
for j=-r:r            % Calculating the positive and negative
   for l=-r:r                                    %neighbourhood effects
      z=ip(rows+k,columns+l)
      if z<i
         nz=nz+(i-z);
         nzc=nzc+1;
      elseif z>=i
         pz=pz+(z-i);
         pzc=pzc+1;
      end
   end
end
pz=pz-i;   % Nullifying self weight
pzc=pzc-1;
mnz=nz/nzc;  % Calculating the mean
mpz=pz/pzc;
if nz>pz     % Determining the direction and final background effects
   dir=1;
   bf=mnz*dir;
elseif nz<=pz
   dir=-1;
   bf=mpz*dir;
end

df=df(i)*dir;
```

Appendix B

In this Appendix, we show the experimental results of the two dimensional clinical ultrasound images. The images used here are taken from GE 700 series machines. These images were converted from the DICOM format to JPEG format before processing. As all the images in Set 1 have a dynamic range of 255, the results produced by the SICNN system would be the same as the results of the SICNN system with dynamic range expansion. In the following experimental outputs, we present the input image and the third iteration SICNN response. The corresponding histograms are also shown to observe the change in the intensity patterns. The contrast scores of the images are also shown. The percentage enhancement is quantified using the relative contrast index.

Image 1:

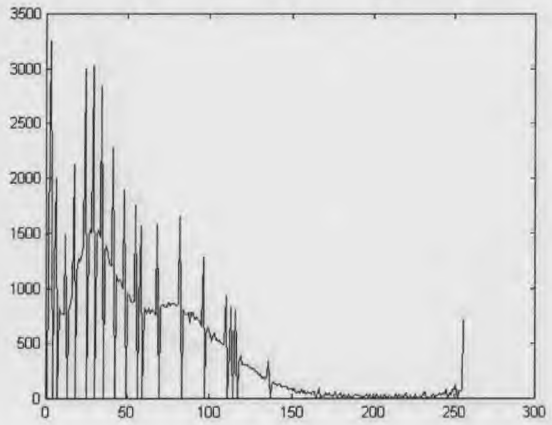


Set 1 Image 1	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	28.8639	-
SICNN output for Iteration 3	255	30.8193	1.2208

***Figure* B.1** (*a*) *Input Image* 1 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 22% *enhancement* (*d*) *Output histogram.*

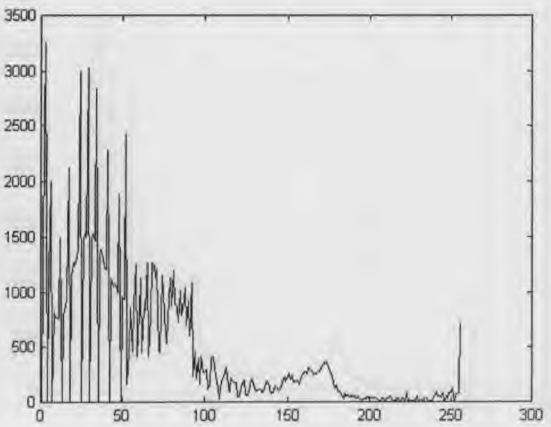
Image **2:**

(*a*)



(*b*)

(*c*)

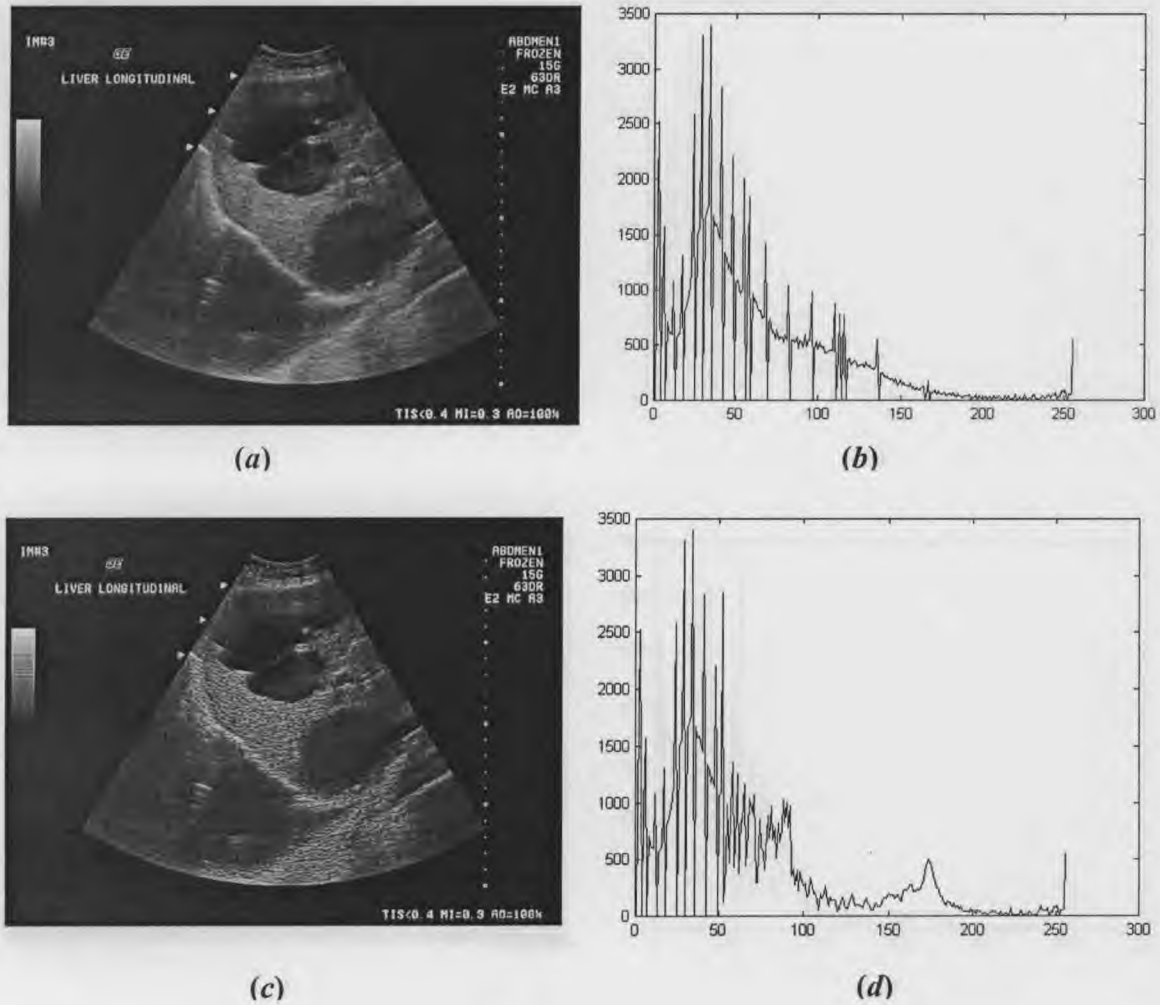


(*d*)

Set 1 Image 2	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	28.4448	-
SICNN output for Iteration 3	255	31.9234	1.2984

***Figure* B.2** (*a*) *Input Image* 2 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 29% *enhancement* (*d*) *Output histogram.*

Image **3:**

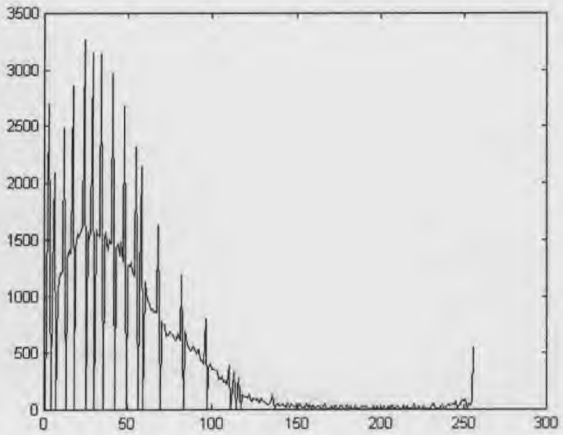


Set 1 Image 3	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	27.0722	-
SICNN output for Iteration 3	255	30.7791	1.3082

***Figure* B.3** (*a*) *Input Image* 3 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 30% *enhancement* (*d*) *Output histogram.*

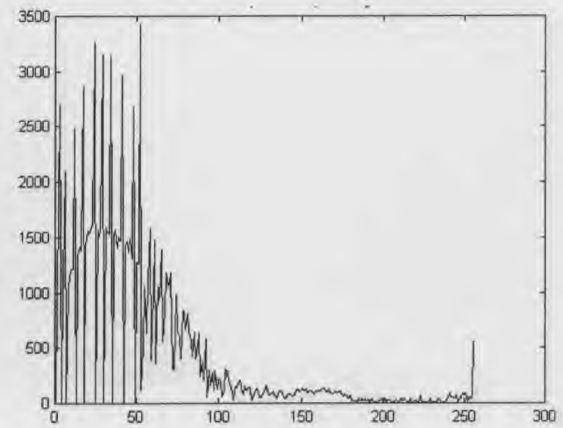
Image **4:**

(*a*)



(*b*)

(*c*)



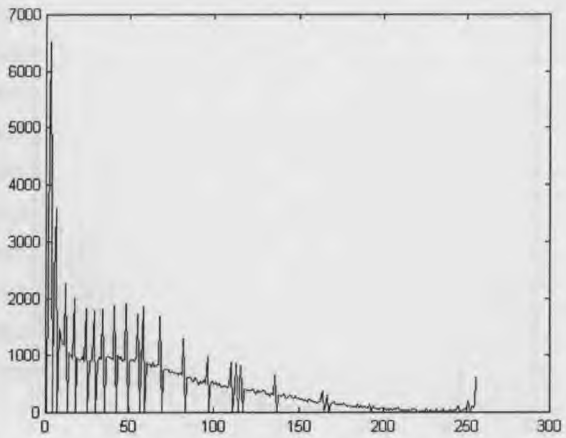
(*d*)

Set 1 Image 4	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	23.5894	-
SICNN output for Iteration 3	255	26.3314	1.1990

***Figure* B.4** (*a*) *Input Image* 4 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 19% *enhancement* (*d*) *Output histogram.*

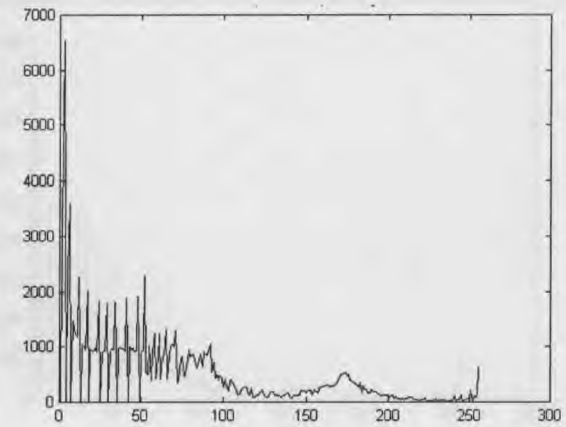
Image **5:**

(*a*)



(*b*)

(*c*)

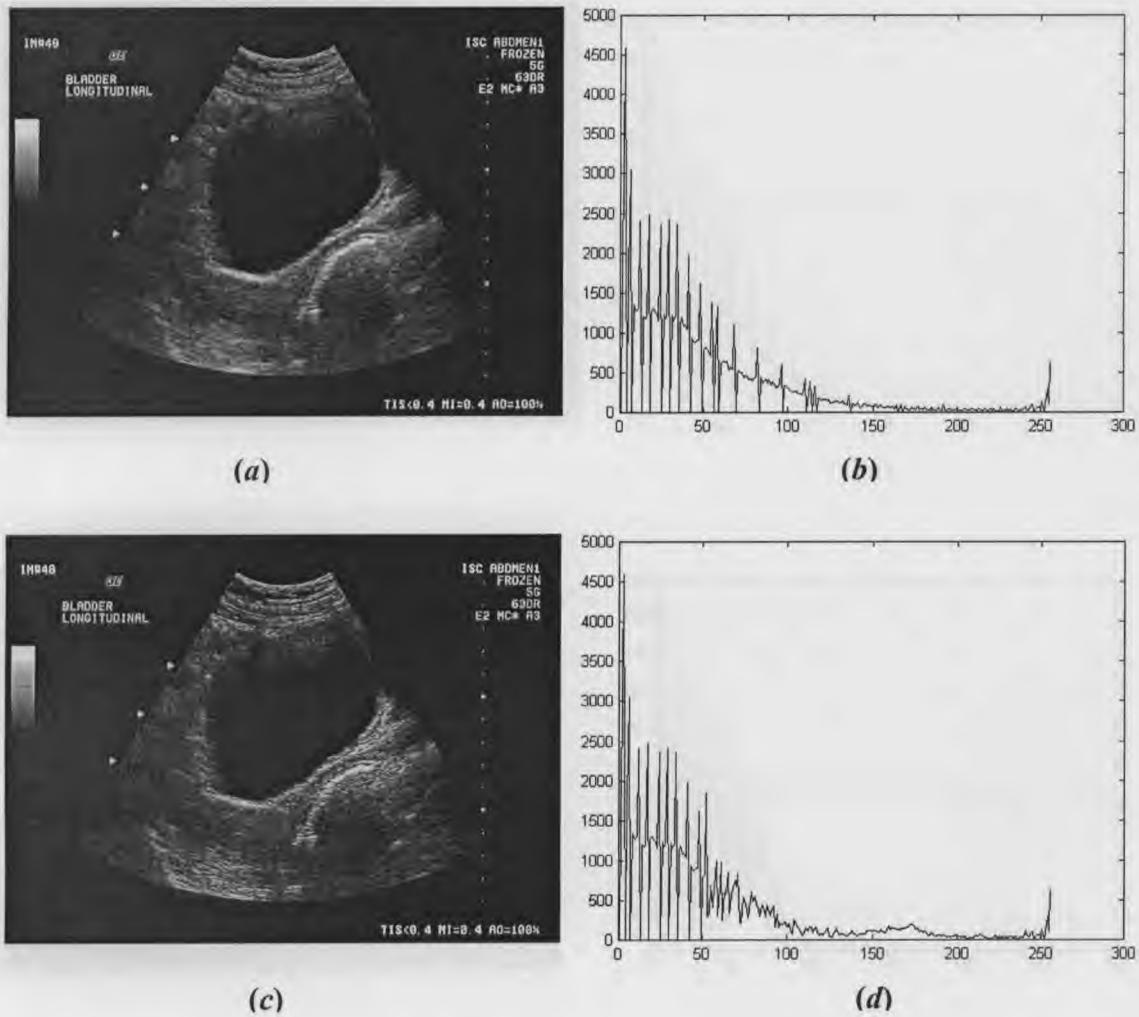


(*d*)

Set 1 Image 5	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	26.2523	-
SICNN output for Iteration 3	255	29.9366	1.3282

***Figure* B.5** (*a*) *Input Image* 5 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 32% *enhancement* (*d*) *Output histogram.*

Image 6:

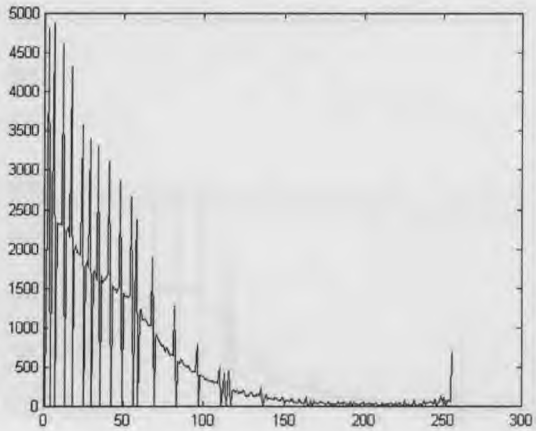


Set 1 Image 6	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	24.5174	-
SICNN output for Iteration 3	255	26.5654	1.2168

***Figure* B.6** (*a*) *Input Image* 6 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 21% *enhancement* (*d*) *Output histogram.*

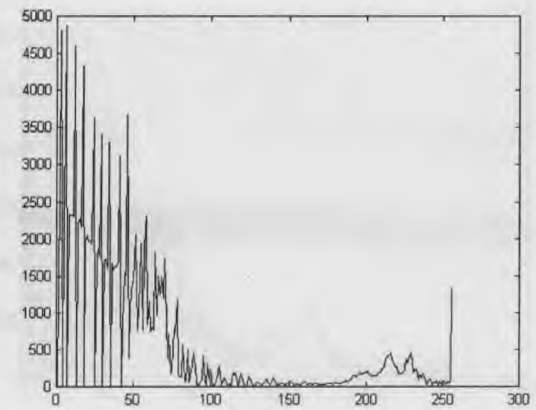
Image 7:

(*a*)



(*b*)

(*c*)



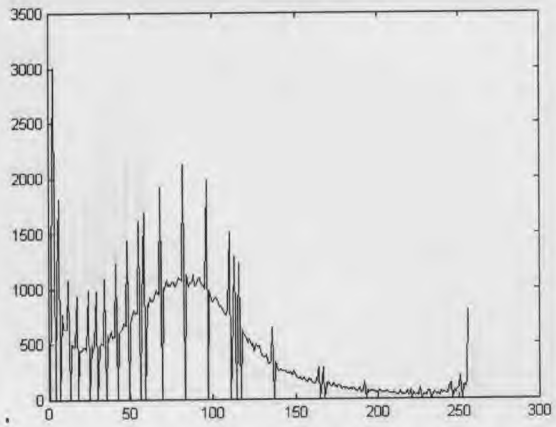
(*d*)

Set 1 Image 7	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	28.1066	-
SICNN output for Iteration 3	255	30.5634	1.2650

***Figure* B.7** (*a*) *Input Image* 7 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 26% *enhancement* (*d*) *Output histogram.*

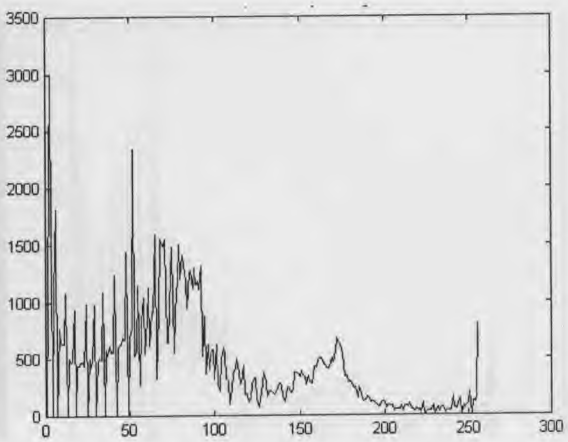
Image **8:**

(*a*)



(*b*)

(*c*)



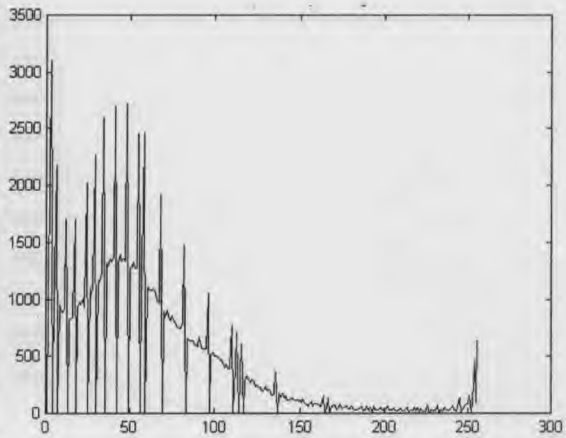
(*d*)

Set 1 Image 8	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	32.9698	-
SICNN output for Iteration 3	255	37.3707	1.3632

***Figure* B.8** (*a*) *Input Image* 8 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 36% *enhancement* (*d*) *Output histogram.*

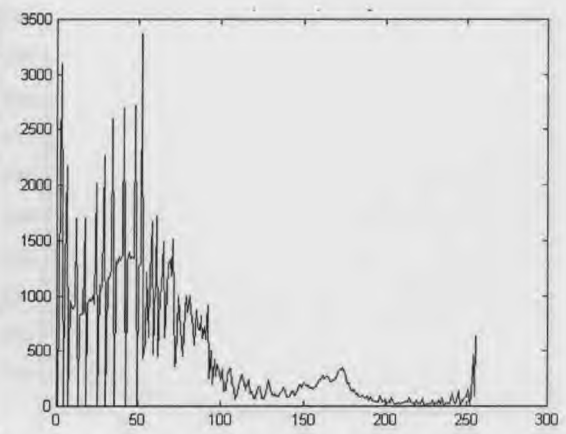
Image 9:

(*a*)



(*b*)

(*c*)

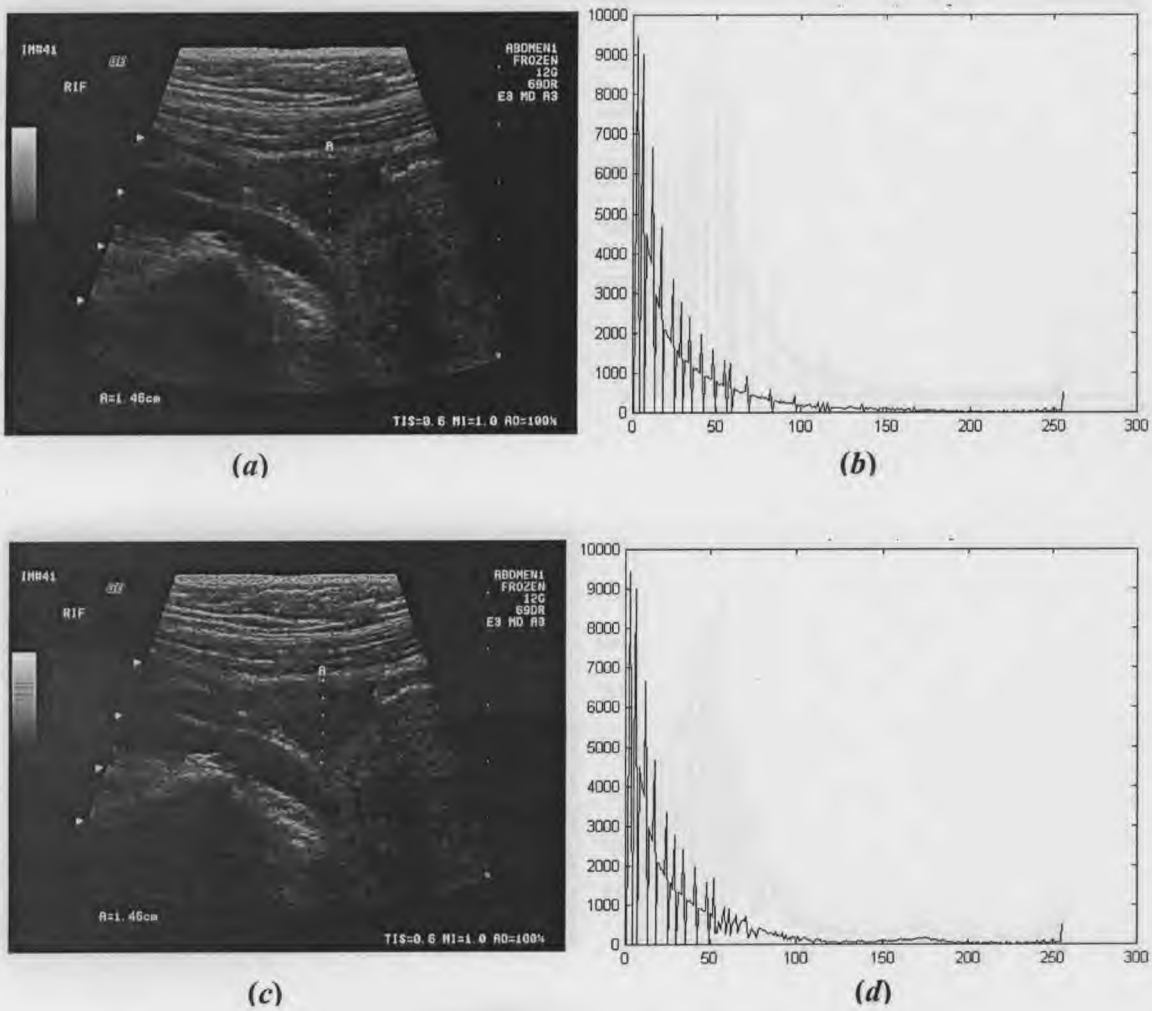


(*d*)

Set 1 Image 9	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	24.2076	-
SICNN output for Iteration 3	255	27.8103	1.2580

***Figure* B.9** (*a*) *Input Image* 9 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 25% *enhancement* (*d*) *Output histogram.*

Image **10:**

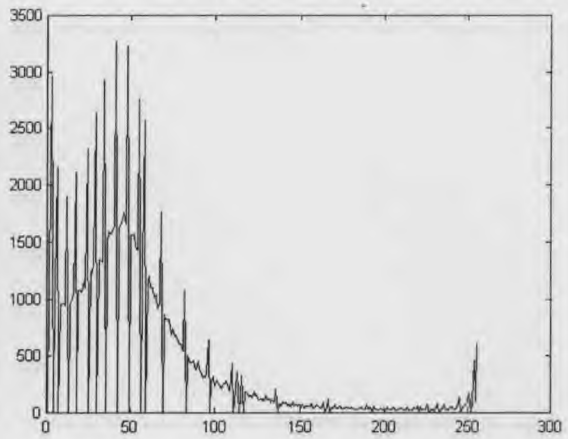


Set 1 Image 10	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	20.1206	-
SICNN output for Iteration 3	255	21.9177	1.1856

***Figure* B.10** (*a*) *Input Image* 10 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 18% *enhancement* (*d*) *Output histogram.*

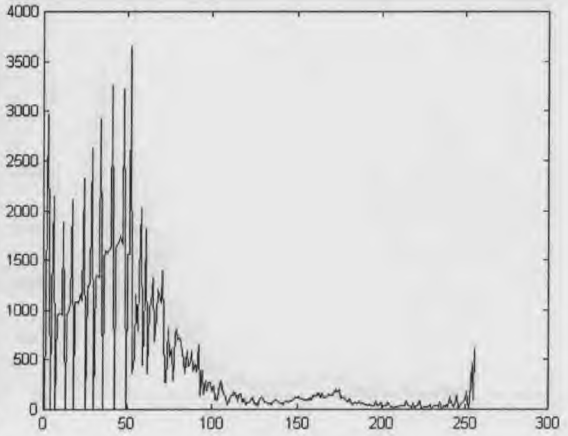
Image **11:**

(a)



(b)

(c)

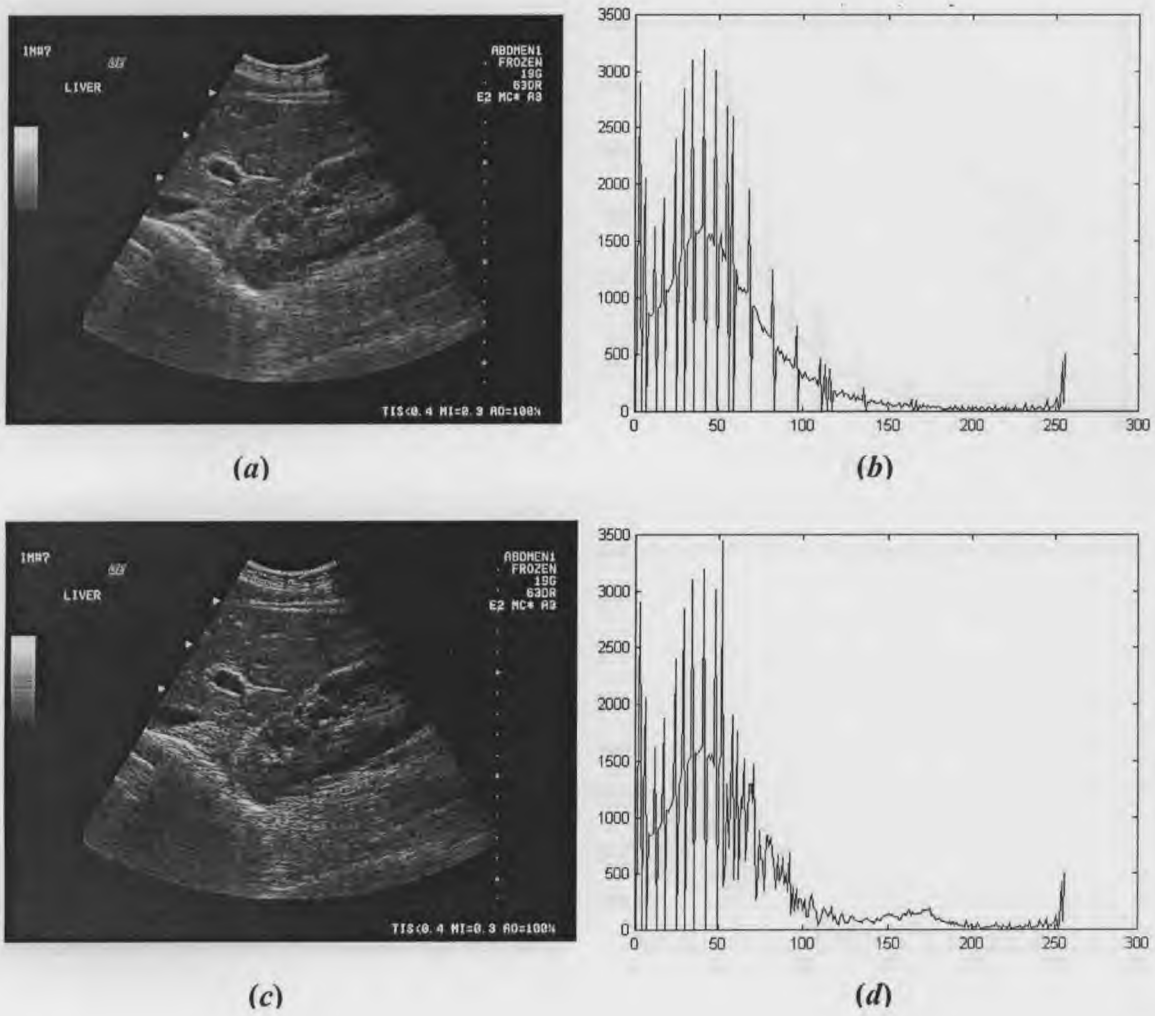


(d)

Set 1 Image 11	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	30.0757	-
SICNN output for Iteration 3	255	32.4573	1.2446

***Figure* B.11** (*a*) *Input Image* 11 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 24% *enhancement* (*d*) *Output histogram.*

Image **12:**

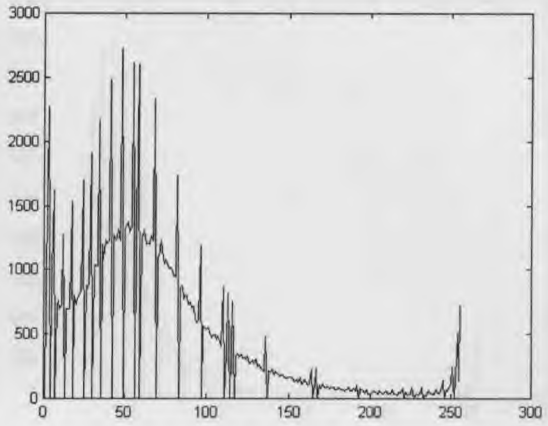


Set 1 Image 12	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	30.7368	-
SICNN output for Iteration 3	255	33.3820	1.2958

***Figure* B.12** (*a*) *Input Image* 12 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 29% *enhancement* (*d*) *Output histogram.*

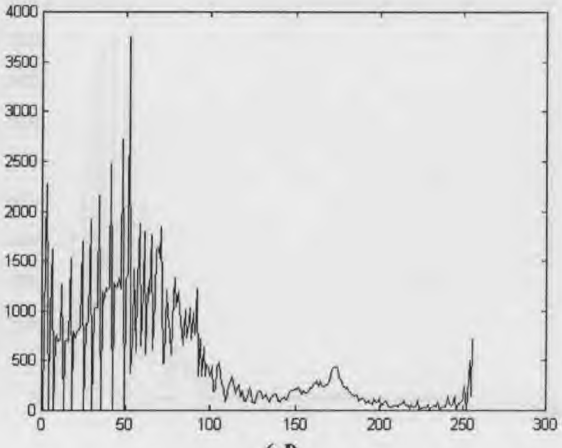
Image **13:**

(*a*)



(*b*)

(*c*)

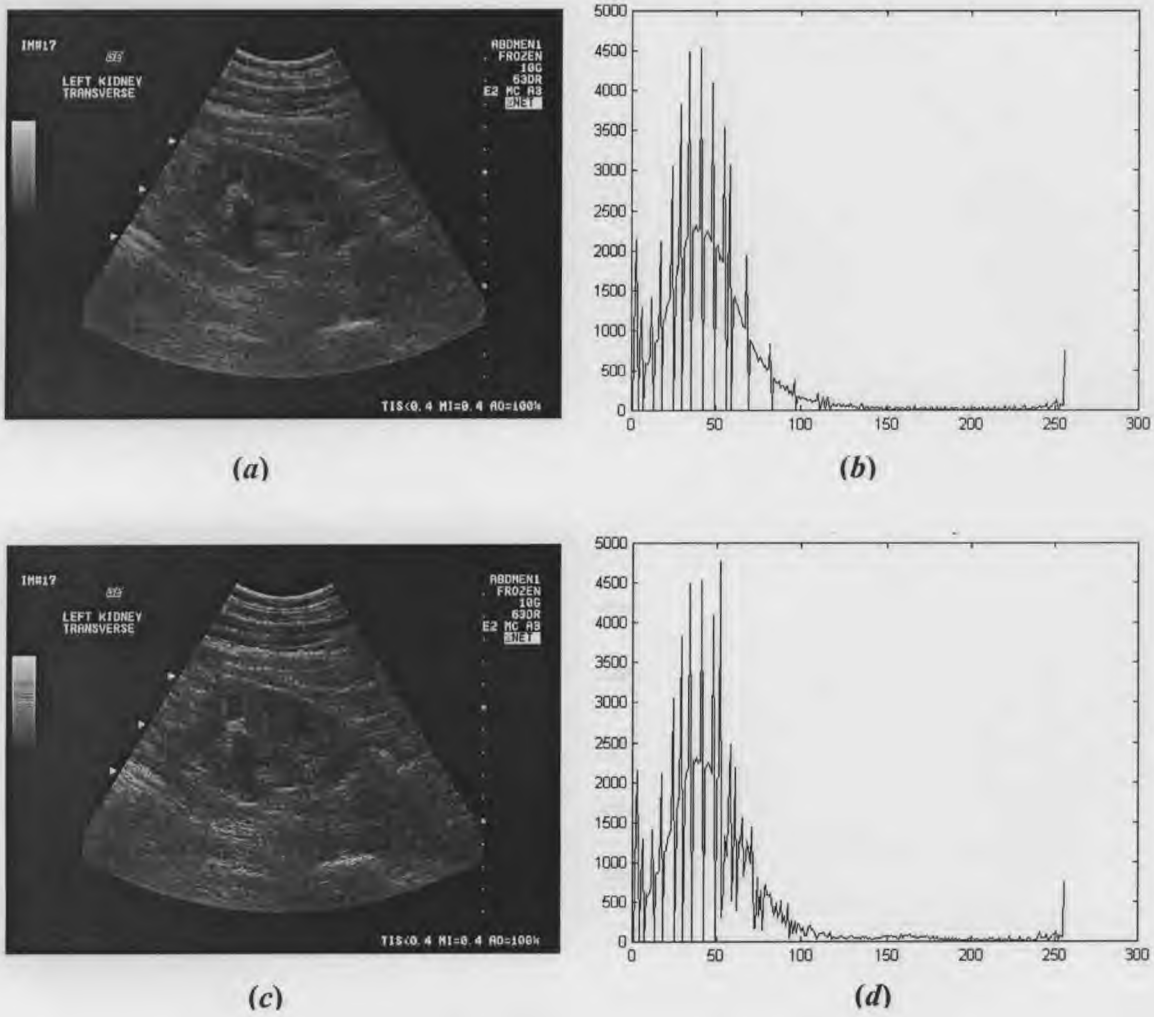


(*d*)

Set 1 Image 13	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	29.6433	-
SICNN output for Iteration 3	255	33.4819	1.3564

***Figure* B.13** (*a*) *Input Image* 13 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 35% *enhancement* (*d*) *Output histogram.*

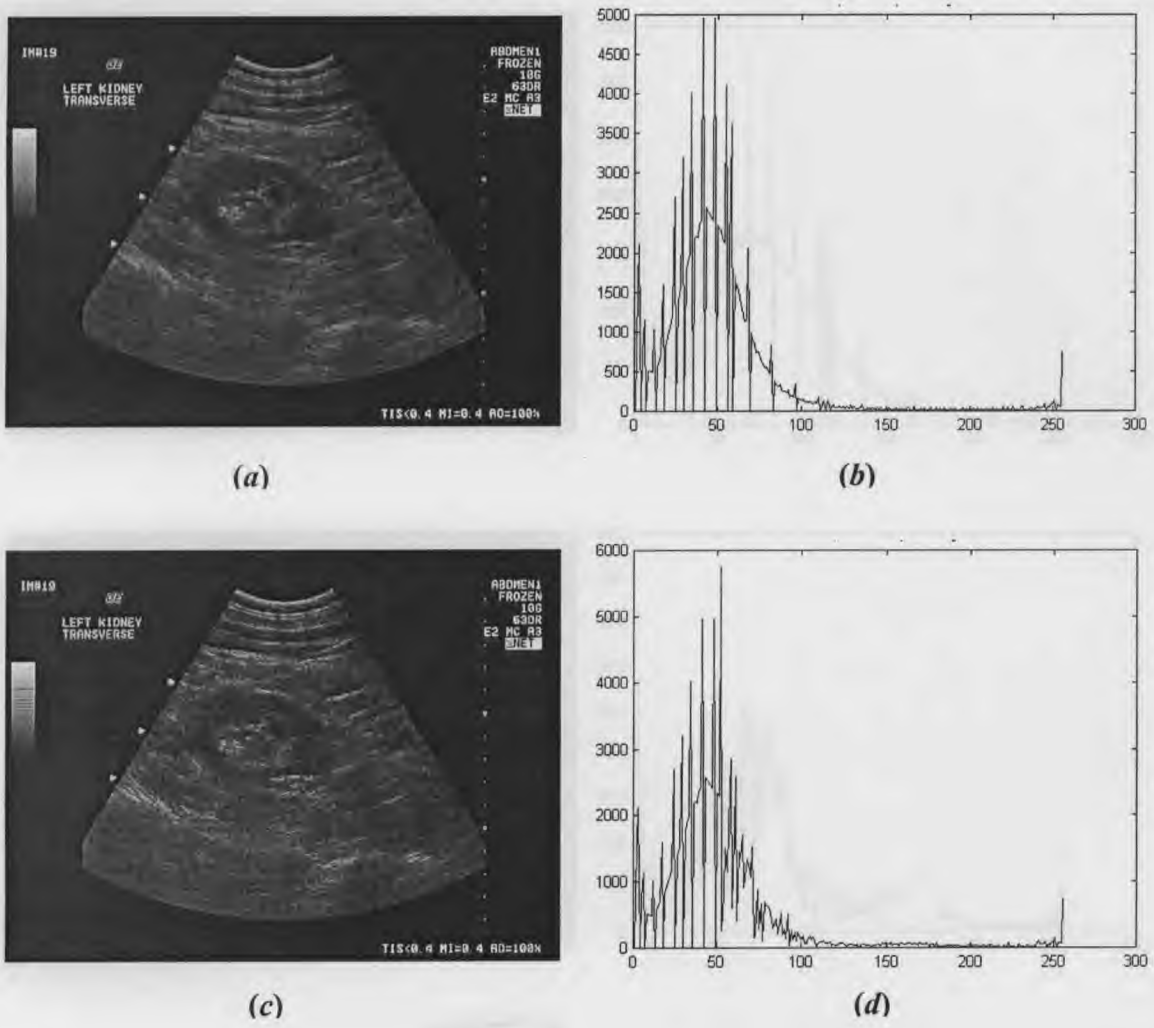
Image **14:**



Set 1 Image 14	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	22.7515	-
SICNN output for Iteration 3	255	23.9882	1.1701

***Figure* B.14** (*a*) *Input Image* 14 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 17% *enhancement* (*d*) *Output histogram.*

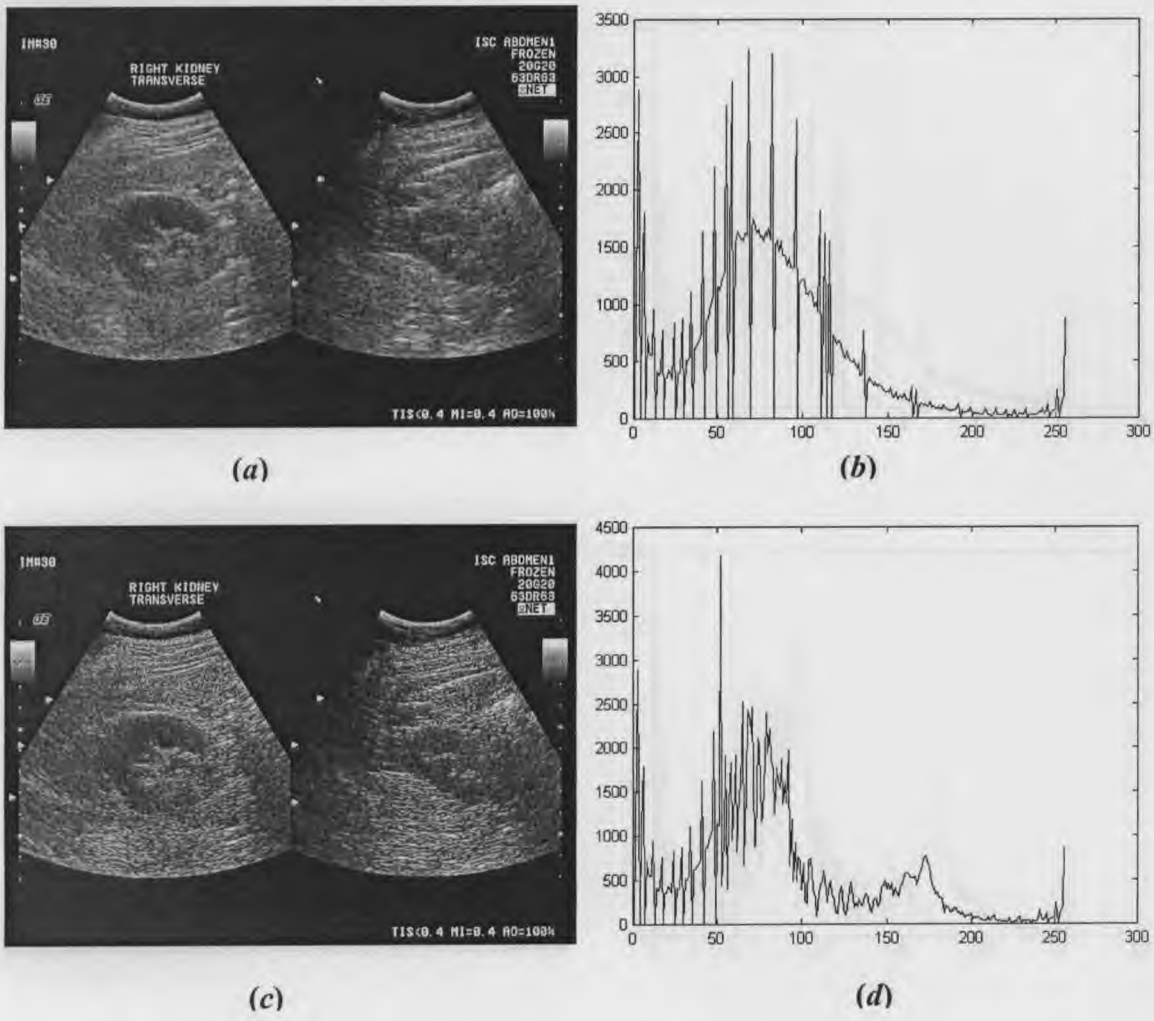
Image **15:**



Set 1 Image 15	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	21.9591	-
SICNN output for Iteration 3	255	23.0607	1.2097

***Figure* B.15** (*a*) *Input Image* 15 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 20% *enhancement* (*d*) *Output histogram.*

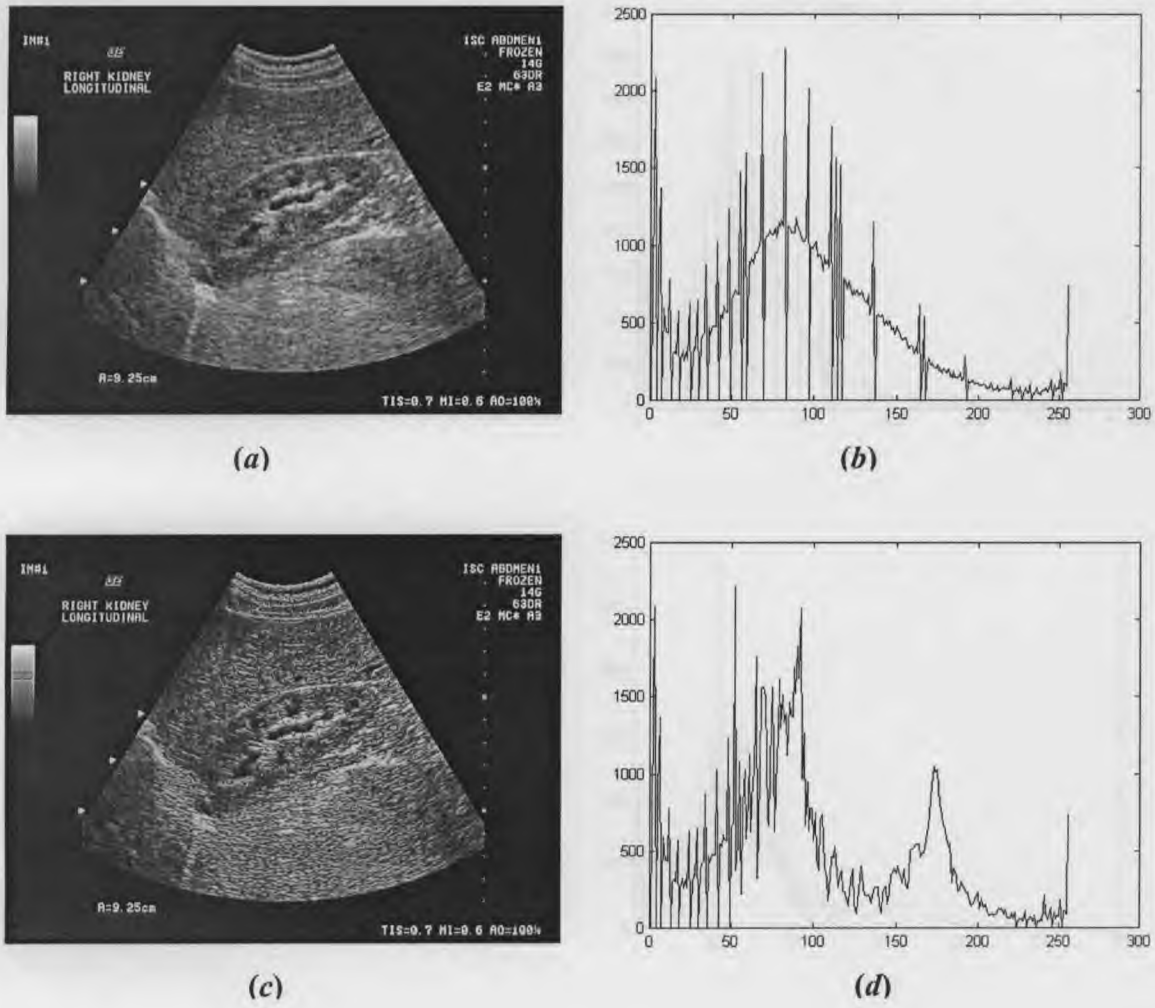
Image 16:



Set 1 Image 16	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	28.1941	-
SICNN output for Iteration 3	255	32.3441	1.3059

***Figure* B.16** *(a) Input Image* 16 *from Set* 1 *(b) Input histogram* (c) *Third iteration SICNN response showing* 30% *enhancement (d) Output histogram.*

Image **17:**

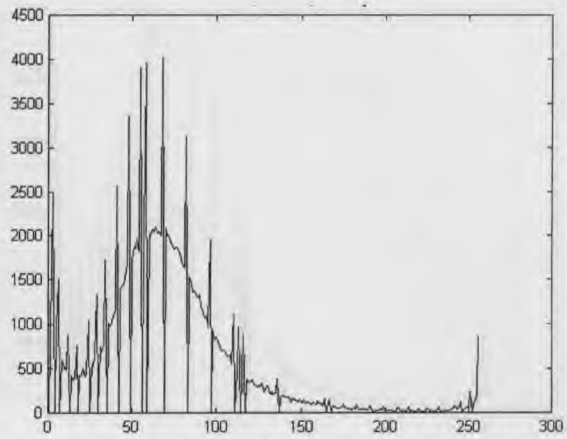


Set 1 Image 17	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	22.0079	-
SICNN output for Iteration 3	255	27.0930	1.3654

***Figure* B.17** (*a*) *Input Image* 17 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 36% *enhancement* (*d*) *Output histogram.*

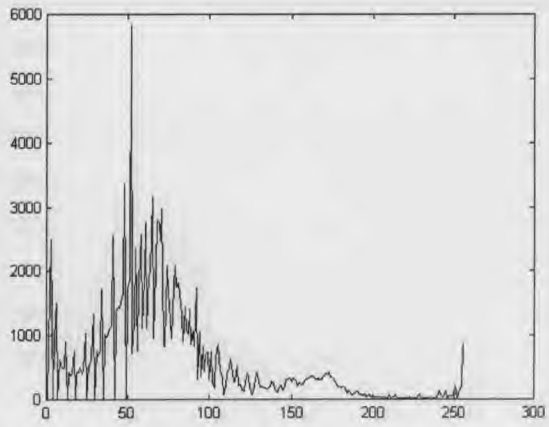
Image **18:**

(*a*)



(*b*)

(*c*)

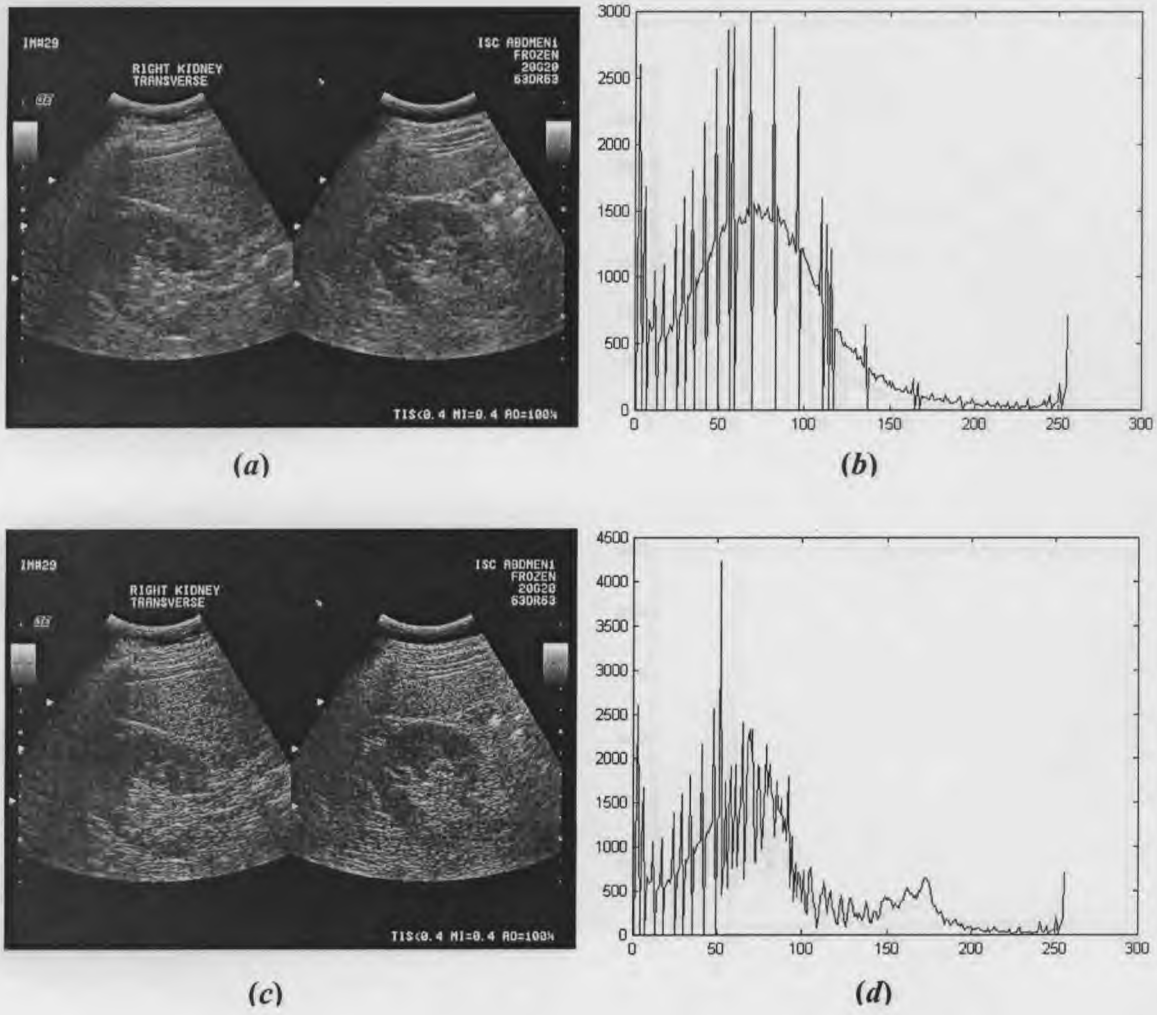


(*d*)

Set 1 Image 18	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	24.3410	-
SICNN output for Iteration 3	255	29.3529	1.3357

***Figure* B.18** (*a*) *Input Image* 18 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 33% *enhancement* (*d*) *Output histogram.*

Image 19:

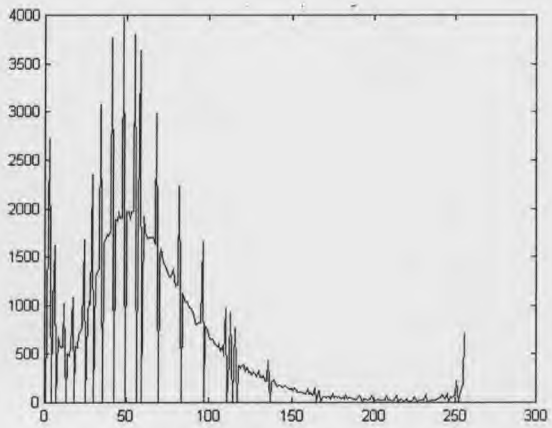


Set 1 Image 19	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	25.3727	-
SICNN output for Iteration 3	255	29.0182	1.3649

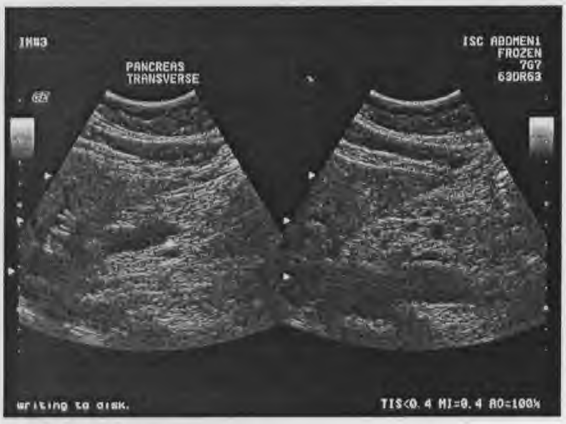
***Figure* B.19** (*a*) *Input Image* 19 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 36% *enhancement* (*d*) *Output histogram.*

Image **20:**

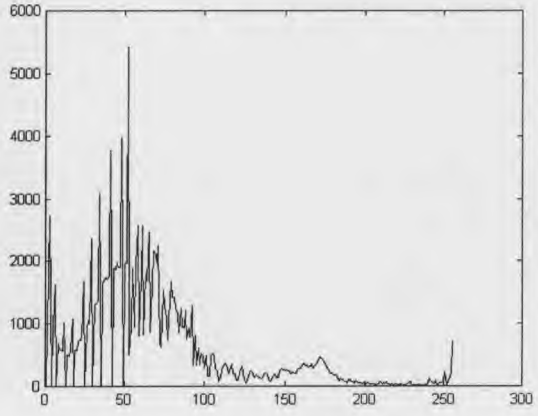
(*a*)



(*b*)



(*c*)



(*d*)

Set 1 Image 20	*Dynamic Range*	*Contrast Score*	*RCI ip-op*
Input Image	255	25.3703	-
SICNN output for Iteration 3	255	28.4183	1.2465

***Figure* B.20** (*a*) *Input Image* 20 *from Set* 1 (*b*) *Input histogram* (c) *Third iteration SICNN response showing* 24% *enhancement* (*d*) *Output histogram.*